AI4SoilHealth

Comparison of pure machine learning and hybrid approaches for integrative soil organic carbon modelling

D3.7

Version 1.1
17th December 2025
Lead Author: **Bernhard Ahrens (Max Planck Institute for Biogeochemistry)**

Contributors: Xuemeng Tian (OpenGeoHub Foundation), Davide Consoli (OpenGeoHub Foundation), Sarem Norouzi (Aarhus University)
Internal Reviewers: Lucas de Carvalho Gomes (Aarhus University), Emmanuel Arthur (Aarhus University)

Action Number: 101086179

Action Acronym: AI4SoilHealth
Action title: Accelerating collection and use of soil health information using AI technology to support the Soil Deal for Europe and the EU Soil Observatory

HISTORY OF CHANGES		
Version	Publication date	Changes
1.0	10/12/2025	• Initial version
1.1	17/12/2025	• Revised version after review

Contents

1 Introduction

The aim of deliverable D3.7 is to compare pure machine learning approaches with novel hybrid approaches for soil organic carbon (SOC) modelling. In soil science, the idea of bringing together the data-adaptiveness of machine learning and process-based understanding has been discussed at least since 2002. At that time, Minasny and McBratney (2002) suggested learning the parameters of a process-based soil water retention curve function with a neural network. Recent technical and methodological advances, namely differentiable programming, have now made it possible to efficiently fuse machine learning with process-based modelling (Reichstein et al., 2019). With differentiable programming the gradient of a loss function with respect to model parameters can be easily calculated which allows the efficient optimization of deep learning models with thousands or even millions of parameters. Minasny and McBratney (2002), for example, had to resort to non-differentiable fine-tuning rather than today's state-of-the-art differentiable modelling to train their hybrid soil-water retention curve model. Since then, hybrid modelling in soil science (Minasny et al., 2024; Tao et al., 2023) and in environmental science in general (Kraft, Jung, Körner, Koirala, & Reichstein, 2021; Reichstein et al., 2019; Tsai et al., 2021) has become one of the fastest-growing fields of research, in part due to the widespread access to differentiable machine learning frameworks such as PyTorch or JAX, and differentiable programming languages such as Julia. Minasny et al. (2024) highlighted that digital soil mapping can benefit from hybrid models, or "soil-science informed machine learning" as they call it. Digital soil mapping is the process of combining field or site measurements of soil variables with spatial covariates to learn the spatial drivers of these soil variables (McBratney, Mendonça Santos, & Minasny, 2003). The trained model can then be used to build soil maps using maps of the spatial covariates.

In this context, digital soil mapping provides a particularly relevant application domain for hybrid models. Digital soil mapping typically operates in a small- to medium-data regime, rather than the "big data" setting of many classical machine-learning applications. In this setting, including sources of knowledge beyond the data itself can increase the robustness and domain of applicability of models. Process-based components can help to extrapolate to new temperature, moisture, or productivity regimes by encoding well-established, theory-based relationships. This is particularly relevant under global change scenarios, where models are routinely applied outside the range of historical observations.

Within carbon-cycle modelling, SOC remains one of the most data-limited components. Vegetation biomass has been successfully derived on large swaths of the earth by using allometry and remote sensing observations that have recently put into question the large, proposed land carbon sink (Bar-On et al., 2025). Eddy covariance flux-towers networks in the EU provide detailed measurements of the carbon balance of ecosystems. Soil organic carbon measurements in the EU benefit from a comparatively good sampling density at the continental scale, thanks to coordinated campaigns such as LUCAS. However, they are still temporally sparse and cannot match the process understanding we obtain from half-hourly eddy-covariance fluxes.

To show what benefits hybrid modelling can bring to field of digital soil mapping, we developed a general-purpose hybrid modelling framework in the programming language Julia. While the focus of this work is hybrid SOC modelling, we designed our hybrid modelling framework EasyHybrid.jl (https://github.com/EarthyScience/EasyHybrid.jl; https://zenodo.org/records/17794983) to be more generally applicable. It can also be linked to other components of the ecosystem carbon balance and digital soil mapping that depend on SOC. For example, it can be used to model how SOC affects soil porosity, bulk density, or soil water retention. Throughout this report, we focus on the potential of hybrid approaches for SOC modelling. Additionally, we show how hybrid modelling can lead to more integrative SOC modelling by linking to other aspects of soil health such as porosity and soil water retention. We start by introducing our design choices for EasyHybrid.jl and show how it can serve as an easy entry point for hybrid models within the digital soil mapping field.

In summary, this deliverable compares pure machine learning and hybrid models for SOC, introduces the EasyHybrid.jl framework, and evaluates how hybrid approaches can support integrative soil health modelling under data limitations and global change.

2 Data

This analysis used only LUCAS soil data from the 2018 campaign restricted to topsoil samples from 0–20 cm depth (Orgiazzi, Ballabio, Panagos, Jones, & Fernández-Ugalde, 2018). For each location we used SOC (g/kg), the volumetric coarse fragment content CF (fraction, unitless, between 0 and 1), and the fine-earth bulk density BD (g/cm³). Where all three variables were available, SOC density (kg/m³) was calculated as:

$$\text{SOC density} = \text{SOC content} \cdot \text{BD} \cdot (1 - \text{CF}).$$

After discarding records that lacked the required covariates, 16,743 of the original 19,036 measurements remained. All 16,743 have SOC density; among these, 5,194 also contain CF and BD. Where BD was not available, we used the bulk density pedotransfer function of Tao et al. (2023). Climatic and remote-sensing covariates followed the covariate stack used within WP5 (Tian, Consoli, et al., 2025; Tian, De Bruin, et al., 2025). In brief, climatic predictors were derived from the CHELSA v2.1 climate time-series (Karger et al., 2017), using a subset of climatic and bioclimatic variables at 1 kilometre-scale resolution. Remote-sensing predictors are based on the Landsat Analysis-Ready Data version 2 (ARD V2) developed by the GLAD group at University of Maryland. For details on preprocessing and variable selection, we refer to the two original papers (Tian, Consoli, et al., 2025; Tian, De Bruin, et al., 2025). Particulate organic carbon (POC) and mineral-associated organic carbon (MOC) data are taken from two studies led or co-led by JRC (Breure et al., 2025; Cotrufo, Ranalli, Haddix, Six, & Lugato, 2019). In these studies, a subset of LUCAS samples was physically fractionated into particulate and mineral-associated fractions, and these measurements were then used to calibrate Vis–NIR spectroscopy models that predict POC and MOC for the wider LUCAS network. Vis–NIR (visible–near infrared) spectroscopy is a non-destructive analytical technique that relates soil reflectance in the visible and near-infrared wavelength range to laboratory-measured soil properties. In our analysis, we combined measured size fractions from the calibration set with Vis–NIR-derived estimates of POC and MOC. Microbial biomass data were from Smith et al. (2021). The POC, MOC and microbial biomass carbon (MBC) data were merged with the original SOC dataset, which did not contain SOC fractions or MBC. For the neural network, all covariates were standardised (z-transformed) to zero mean and unit variance before training. The final covariate set consisted of 168 individual covariates (we did not use any of the categorical variables from the original stack).

3 Pure machine-learning benchmark for soil organic carbon, carbon fractions and microbial biomass carbon

3.1 Neural network architecture for multi-task SOC modelling

Figure 1 shows a classical machine-learning architecture for a multi-task problem. Predictors X are the inputs to a neural network that has four output nodes for SOC, MOC, POC, and MBC. In digital soil mapping, regression problems are often set up with single-output algorithms, typically random forests, although multi-output versions exist. One could hypothesize that multi-output setups achieve better performance, since relationships between different outputs can be shared. In reality, however, very few studies in digital soil mapping use multi-output algorithms.

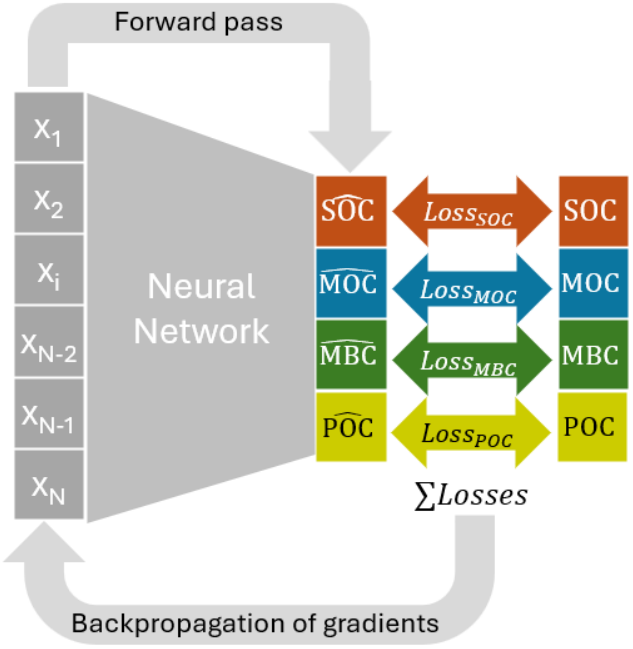

Figure 1 Conventional neural network with x_i covariates/features as input and four output nodes – predicted $\widehat{\mathrm{SOC}}$ (soil organic carbon), $\widehat{\mathrm{MOC}}$ (mineral-associated organic carbon), $\widehat{\mathrm{POC}}$ (particulate organic carbon), and $\widehat{\mathrm{MBC}}$ (microbial biomass carbon). The loss function aggregates the discrepancies between predictions and observations. The gradients of the loss with respect to the parameters are passed backwards through the neural network, telling the optimizer how to adjust each parameter.

Formally, we denote the neural network by θ_{NN} and write

$$\hat{y} = \theta_{\text{NN}}(X),$$

where $\hat{y}$ is a vector with four components (SOC, MOC, POC, MBC). Neural networks can be viewed as a sequence of layers that are chained together, for example a chain of dense layers that form a classical multilayer perceptron:

$$\hat{y} = \theta_{\text{NN}}(X) = \text{Chain}(D_1, \dots, D_N),$$

where $D_1, \dots, D_N$ are N dense layers. In the pure machine-learning benchmark, the last dense layer directly outputs the four target variables.

In this deliverable, the multi-task neural network serves as the pure machine-learning benchmark against which we compare the hybrid model (section 4).

3.2 Loss function: KGE-based multi-task learning

Critical for multi-task learning is the definition of the loss function, since multiple output losses must be combined into one composite loss. In this report, we used $1 - \text{KGE}$, the Kling–Gupta efficiency, as the loss function for SOC, MOC, POC, and MBC. The KGE combines correlation, bias and variability into a single performance metric and is commonly used in hydrological modelling (Gupta, Kling, Yilmaz, & Martinez, 2009). We define KGE as

$$\text{KGE} = 1 - \sqrt{(r-1)^2 + (\alpha-1)^2 + (\beta-1)^2},$$

where r is the Pearson correlation coefficient between observed and predicted values, $\alpha = \sigma_{\text{pred}}/\sigma_{\text{obs}}$ is the ratio of standard deviations, and $\beta = \mu_{\text{pred}}/\mu_{\text{obs}}$ is the ratio of means. KGE thus uses the Euclidean distance from the ideal point to summarise three aspects of model performance: how well temporal or spatial patterns (here: spatial) are represented (via r), the bias of the model (via β), and how well the variability of the dataset is matched (via α). The KGE-based loss has the convenient property that it is dimensionless.

Concretely, we computed KGE separately for SOC, MOC, POC, and MBC, convert each to a loss via $1 - \mathrm{KGE}$, and minimised the mean of these four losses during training. This has the additional benefit that the composite loss remains interpretable, so that one can judge at a glance how well the model is performing.

3.3 Multi-task neural network for SOC, MOC, POC and MBC

As a pure machine-learning baseline, we implemented a multi-task neural network that predicts four soil variables simultaneously: SOC, MOC, POC and MBC. The network takes a set of predictor variables (see section 2) as input and outputs the four targets in a single forward pass.

The network architecture consisted of an input normalisation, followed by three hidden layers with decreasing width and dropout for regularisation. Concretely, we used fully connected layers with 256, 128, 64 and 32 units, each followed by a sigmoid activation, and apply a dropout rate of 0.3 after the first three hidden layers. A final dense layer mapped from 32 hidden units to the four output nodes (SOC, MOC, POC, MBC). No explicit global or mechanistic parameters were used; all learnable parameters belonged to the neural network.

The model was trained using the RMSProp optimiser with a learning rate of 0.01, a batch size of 2048, and a maximum of 1000 epochs. We employed early stopping with a patience of 100 epochs. As loss function we used the KGE-based multi-task loss described in section 2.2: for each target variable, the Kling–Gupta efficiency (KGE) was computed, converted to a loss via $1 - \mathrm{KGE}$, and the mean loss across the four targets was minimised. During training, observations were shuffled, and additional diagnostic metrics (including α, β, and Pearson correlation reported in Figure 2) were tracked to characterise model performance.

In Figure 2, we show scatterplots of the performance for the training (train) and validation (val) sets. We report performance for both, even though we are primarily interested in validation performance; here, profiles are split into 80% for training and 20% for validation/test. We focus on the discrepancy between training and validation metrics to check whether some datasets suffer from overfitting, especially datasets with fewer observations. However, we see that the largest discrepancy between training KGE (0.89) and validation KGE (0.25) exists for the data stream with the fewest points, MBC. The neural network was clearly overfitted on this small dataset of N = 331 that the optimizer sees more often than the large SOC training dataset with N = 11610 (KGE_train = 0.74 and KGE_val = 0.6). Model performance for the SOC fractions MOC and POC lay between

these two extremes. Overall, this highlights the problems of multi-task learning on data streams with different numbers of observations. More advanced techniques such as Kendall's uncertainty weighting between data streams may remedy this problem (Kendall, Gal, & Cipolla, 2018).

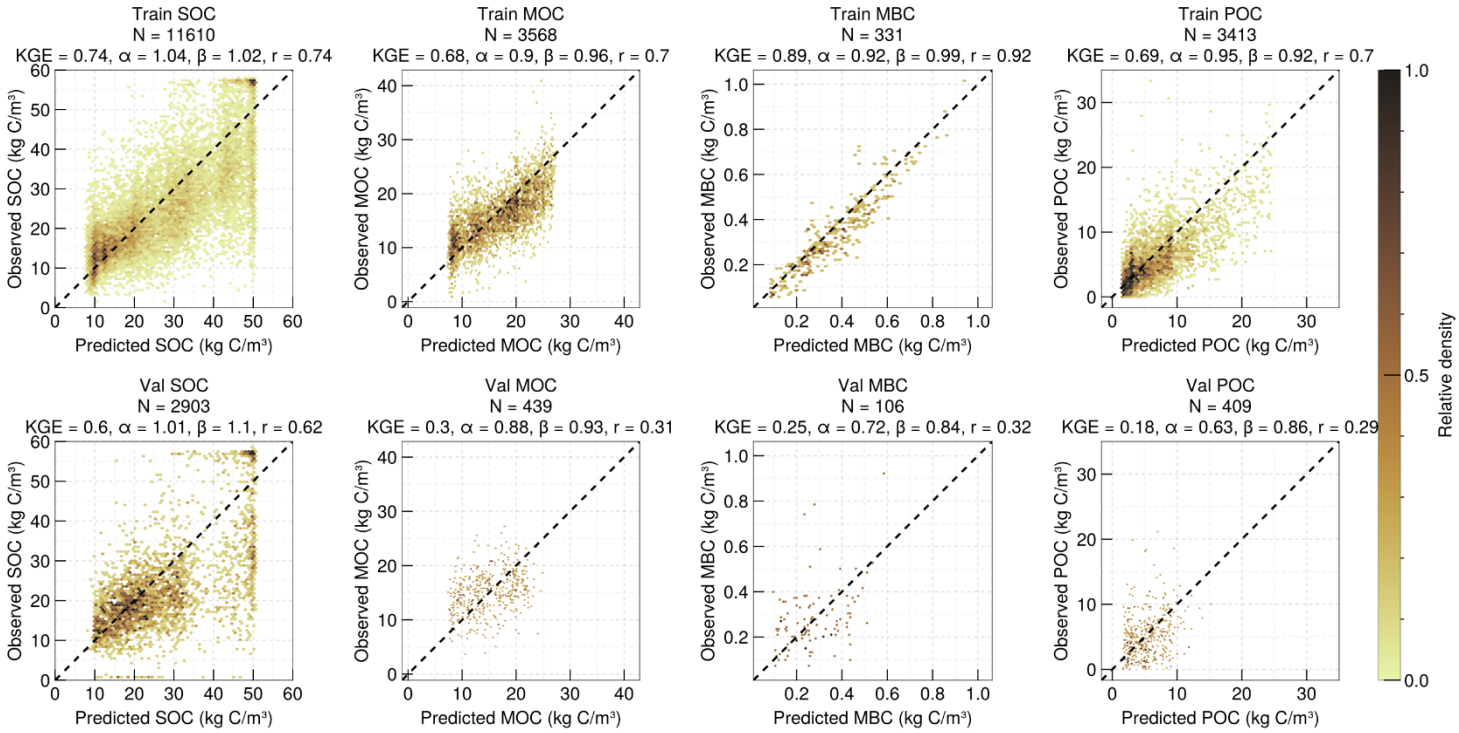

Figure 2. Training and validation for a pure neural network in a multi-task setup. Performance of the hybrid model in training (upper) and validation (lower) for SOC, MOC, MBC and POC. KGE (Kling-Gupta efficiency) measures model performance as the Euclidean distance of three components - correlation r, variability ratio α, and bias ratio β - from the ideal point.

4 Hybrid soil organic carbon model and EasyHybrid.jl framework

4.1 Motivation and previous work (Tao et al., 2023)

Tao et al. (2023) is one of the more prominent recent examples of progress towards integrative hybrid modelling of SOC. In that publication, the authors calibrated the parameters of a microbial-explicit SOC model profile by profile from the World Soil Information Service database (Batjes, Ribeiro, & Van Oostrum, 2020). These profile-by-profile parameters were then learned with a neural network for global upscaling.

This approach had a couple of shortcomings: (1) due to the profile-by-profile calibration, the process-based model was generally overparameterized; (2) the upscaling of the profile-by-profile parameters was consequently an upscaling of under-constrained parameters; and (3) the mechanistic model was flawed insofar as the major pool in the model was not dependent on litter inputs in its steady state (He et al., 2024).

Using Tao et al. (2023) as the state-of-the-art starting point, we improved both the mechanistic SOC model and the parameter-learning strategy. These changes result in an end-to-end hybrid model in our implementation.

4.2 Process-based SOC model and temperature sensitivity

Figure 3 illustrates the mechanistic model we implemented and tested as the last layer in the hybrid model. The model is based on the formulation by K. Georgiou, Abramoff, Harte, Riley, and Torn (2017) and is, in essence, also the pool structure that was used in Tao et al. (2023). However, we used a modified version that includes two key changes: (1) we changed the microbial turnover term from a linear dependence on microbial biomass ($k \cdot \mathrm{MBC}$) to a density-dependent, quadratic form ($k \cdot \mathrm{MBC}^2$), restoring the dependence of carbon stocks on litter inputs in steady state (He et al., 2024). (2) Compared to Tao et al. (2023), we also used the K. Georgiou et al. (2017) formulation for the formation of MOC via a Langmuir sorption approach. The Langmuir formulation has the property that it defines an upper limit, a capacity for MOC formation, $Q_{\max}$. This formulation was chosen as an attempt to derive $Q_{\max}$ values equivalent to the ones calculated empirically via boundary/quantile regression (Beare et al., 2014; Feng, Plante, & Six, 2013; Katerina Georgiou et

al., 2022). This can also be seen as a test of whether we need an explicit $Q_{\max}$ formulation in process-based models. It also helps to assess the importance of this parameter.

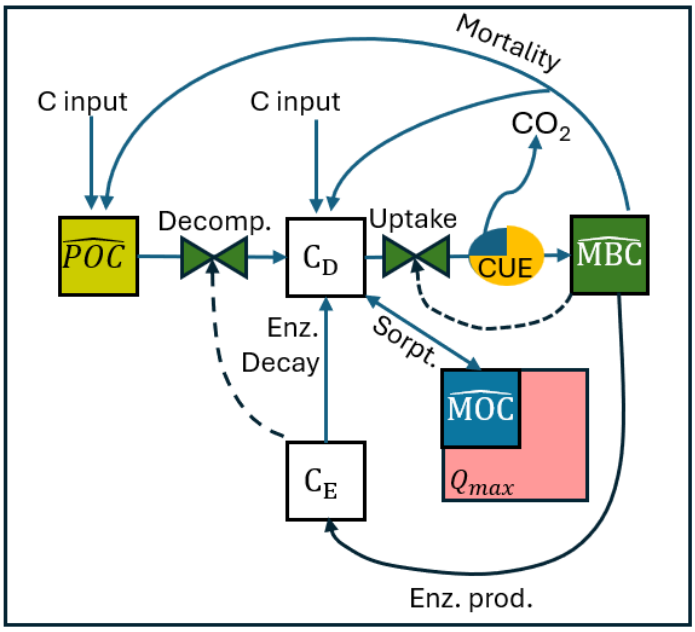

Figure 3: Five-pool microbial model with formation of mineral-associated organic carbon $\widehat{\mathrm{MOC}}$ on mineral surfaces that have a maximum sorption capacity Q_{max} (K. Georgiou et al., 2017). Carbon enters the particulate organic carbon pool $\widehat{\mathrm{POC}}$ and the DOC pool C_D in the form of litter. The $\widehat{\mathrm{POC}}$ pool is decomposed by extracellular enzymes C_E that are produced by microbial biomass $\widehat{\mathrm{MBC}}$. Microbial biomass takes up carbon for growth with carbon use efficiency CUE, while $(1 - \mathrm{CUE}) \times \mathrm{Uptake}$ constitutes what is lost as CO_2 through microbial respiration.

To represent the temperature sensitivity of SOC processes, we used Q_{10} functions. We prescribed Q_{10} values that are based on temperature sensitivities from enzymatic assays for the depolymerization of organic matter in soil and Q_{10} values from isotherm sorption studies in the lab (Ahrens et al., 2020; Wang, Post, Mayes, Frerichs, & Sindhu, 2012). All microbial mediated processes in the model, such as depolymerization of litter, had Q_{10}s in the range of 1.98 to 2.16:

$$Depolymerization = V_{max,\,Tref} \cdot Q_{10,Vmax}^{(T-T_{ref})/10} \cdot MBC \cdot \frac{POC}{KM + POC}$$

Here, $V_{\max,T_{\mathrm{ref}}}$ is the depolymerisation rate at the reference temperature T_{ref}. The term $Q_{10,V_{\max}}^{(T-T_{\mathrm{ref}})/10}$ scales this rate with temperature T, such that a 10 °C increase multiplies the rate by $Q_{10,V_{\max}}$.

$\frac{POC}{KM+POC}$ describes substrate limitation of depolymerization with the Michaelis-Menten or half-saturation constant KM

The Q_{10} values of processes related to the microbial decomposition of organic matter are generally higher, with values around 2, while the adsorption and desorption processes were, respectively, assigned a temperature sensitivity of $Q_{10,ads}$ = 1.08 and $Q_{10,des}$ = 1.34:

$$Sorption = k_{ads,Tref} \cdot Q_{10,ads}^{\frac{T-T_{ref}}{10}} \cdot C_D \cdot \left(1 - \frac{MOC}{Q_{max}}\right) - k_{des,Tref} \cdot Q_{10,des}^{\frac{T-T_{ref}}{10}} \cdot MOC$$

Where $k_{ads,Tref}$ and $k_{des,Tref}$ are the adsorption and desorption rates at reference temperature T_{ref}. $\left(1 - \frac{MOC}{Q_{max}}\right)$ describes the amount of free sorption sites for dissolved organic carbon (C_D)

4.3 EasyHybrid.jl: parameter roles and workflow

We set up EasyHybrid.jl so that users can start from simple definitions of process-based models. The user can then decide which parameters are spatially or temporally varying, which parameters are global in space and time, and which parameters are fixed from other parameterization sources (e.g. literature values, previous calibrations).

In our notation:

- θ_{global}: global parameters, shared across all sites and time;
- $\theta_{NN}(X)$: parameters predicted by a neural network as functions of features/covariates X, and thus allowed to vary in space and/or time;
- θ_{fixed}: fixed parameters taken from independent measurements, literature, empirical relationships or previous calibrations.

EasyHybrid.jl provides a generic way to declare these three parameter types and to train them jointly in an end-to-end differentiable manner. Conceptually, hybrid modelling can be explained from two sides: (i) as a process-based model with machine learning embedded, or (ii) as a machine-learning model with a process-based model as the final layer. In the next subsection, we take a process-based perspective: the hybrid model is a process-based model with machine learning embedded to represent specific uncertain or unresolved processes/parameters.

4.4 Hybrid model from the process-based perspective

We start from a purely process-based model

$$\hat{y} = M(f, \theta_{\text{global}}, \theta_{\text{fixed}}),$$

where M is our SOC model, f represents optional forcing (e.g. temperature or carbon input through litter), and θ_{global} and θ_{fixed} are fixed parameters defined in the notation above.

The innovation of hybrid models lies in the realisation that for some parameters θ it is beneficial to make them depend on temporally and/or spatially varying features. In the past, spatially varying parameters such as the theoretical maximum capacity for MOC formation have typically been derived from quantile regression approaches and therefore calibrated outside the process-based models in which they are later used.

By using differentiable programming languages such as Julia, one can instead embed a neural network directly into a process-based model and still rely on the optimisation techniques and data-adaptiveness that have made machine learning so successful (Innes et al., 2019). Concretely, we implement a neural network $\theta_{\text{NN}}(X)$ that predicts selected parameters as a function of features/covariates X that may be spatially or temporally varying. This leads to the following definition of a hybrid model from the process-based perspective:

$$\hat{y} = M\left(f, \theta_{\text{global}}, \theta_{\text{fixed}}, \theta_{\text{NN}}(X)\right),$$

where parameters that were formerly fixed or globally estimated are replaced by a neural network embedded within the mechanistic model. The network maps covariates to mechanistic parameter values, which are subsequently used in the mechanistic model. In the next subsection, we motivate hybrid modelling from the machine learning perspective. There, we explain how the gradient of the loss between modelled and observed values is used to jointly optimize the neural-network parameters and the globally estimated parameters.

4.5 Hybrid model from the machine-learning perspective

From the machine-learning perspective, the hybrid model can be viewed as a modification of the pure neural network introduced in Section 2.1. Rather than predicting SOC, MOC, POC and MBC directly in the last dense layer,

$$\hat{y} = \theta_{\text{NN}}(X) = \text{Chain}(D_1, \ldots, D_N),$$

we let the neural network predict the parameters of the mechanistic model as latent variables and append the process-based model M as the final layer (Figure 4).

Concretely, the last trainable layer of the neural network outputs a vector of mechanistic parameters θ_M, and the hybrid model can be written as

$$\hat{y} = \text{Chain}(\theta_{\text{NN}}(X) \rightarrow \theta_M \rightarrow M(f, \theta_M)),$$

where θ_M denotes the mechanistic model parameters that are treated as latent outputs of the neural network and are subsequently passed into the process-based model M together with external forcing f.

In addition, there can be global parameters that are learned as a single value across all sites and known constants or parameters can be prescribed as fixed parameters. To incorporate information about mechanistically meaningful bounds for parameters, we use a sigmoid activation function in the last neural-network layer (which corresponds to the second-last layer of the hybrid model). The sigmoid function produces outputs in $[0,1]$, which we then rescale to parameter-specific ranges, e.g.

$$\theta = \theta_{\min} + s\,(\theta_{\max} - \theta_{\min}),$$

where $s \in [0,1]$ is the sigmoid output. This makes it straightforward in EasyHybrid.jl to constrain parameters to biogeochemically plausible ranges.

The key innovation that enabled both deep learning and hybrid modelling is also illustrated in Figure 1 and Figure 4. After the first forward pass through the network and the mechanistic model, we compute a loss that quantifies the discrepancy between model outputs and observations (here, the KGE-based loss described in Section 2.2). With differentiable programming, we can calculate the gradient of this loss with respect to all parameters. These gradients then inform the optimisation algorithm in which direction and by how much the parameters should be changed to minimise the loss function.

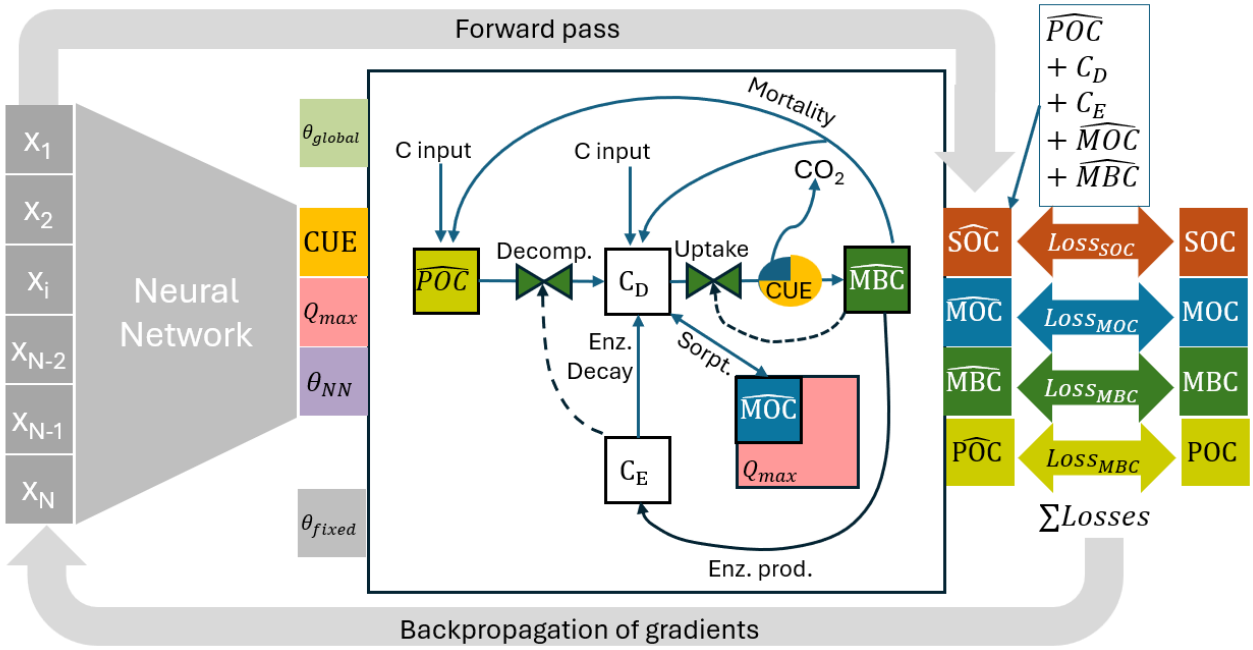

Figure 4. Hybrid model with a neural network with x_i covariates/features as input and three exemplary output nodes – parameters of the mechanistic model, CUE (carbon use efficiency), Q_{max} (theoretical maximum capacity for MOC formation), and other mechanistic parameters θ_{NN}. Together with fixed parameters θ_{fixed} and global parameters θ_{global}, all of these parameters enter the process-based model which predicts $\widehat{\mathrm{SOC}}$, $\widehat{\mathrm{MOC}}$, $\widehat{\mathrm{POC}}$ and $\widehat{\mathrm{MBC}}$. The KGE loss function aggregates the discrepancies between predicted and observed values of the four targets. The gradients of this loss with respect to the mechanistic parameters θ_{global} and θ_{NN} (including CUE and Q_{max}) are passed backwards through the mechanistic model. The gradients with respect to θ_{NN} are further passed backwards through the neural network. The optimizer adjusts the parameters of the neural network, the mechanistic parameters predicted from the neural network θ_{NN} and the global parameters θ_{global}.

4.6 Hybrid multi-task learning for multiple soil health indicators

The neural network of the hybrid model saw the same predictor set X as the pure machine-learning baseline (Section 3.1), but instead of directly predicting SOC, MOC, POC and MBC, it predicts selected parameters of the mechanistic model. The process-based model then uses these parameters together with external forcing variables to simulate the target variables.

For the hybrid experiments presented here, we used the Five-pool SOC model (Section 3.5) by K. Georgiou et al. (2017) as the mechanistic component, with mean net primary production (NPP) and

mean annual air temperature (T) as forcing variables. The model was calibrated against four targets: bulk SOC, MOC, POC, and MBC.

The neural-network backbone was identical to the pure NN baseline (Section 3.1): an input normalisation layer followed by three fully connected layers with 256, 128, 64 and 32 units, respectively, each followed by a sigmoid activation and a dropout rate of 0.3. As before, output activations were passed through a sigmoid and linearly rescaled to parameter-specific bounds, ensuring that all predicted parameters remain within biogeochemically plausible ranges.

To investigate how much spatial flexibility is needed in the mechanistic parameters, we systematically varied which parameters are treated as neural-network outputs and which remain global. We considered three configurations:

- **All global**: all parameters of the mechanistic model were calibrated as global parameters (no spatial variation in θ);
- **CUE, $Q_{\max}$, f, $V_{\max,0}$, $K_{M,0}$ neural**: in this configuration, the neural network learns carbon use efficiency (CUE), the maximum sorption capacity $Q_{\max}$, and litter-quality-related parameters. Specifically, it predicts the litter allocation fraction f (how much litter enters the POC versus the DOC pool) and the baseline depolymerisation parameters $V_{\max,0}$ and $K_{M,0}$, while the remaining parameters are kept global.
- **All neural**: all mechanistic parameters were predicted as spatially varying functions of X.

For each configuration, we split the data by site identifier into training and validation subsets to ensure that evaluation was performed on sites not seen during training. The hybrid model was then trained with the same optimisation setup as the pure neural network: RMSProp with a learning rate of 0.01, batch size 2048, up to 1000 epochs, early stopping with a patience of 100 epochs, and the KGE-based multi-task loss described in Section 2.2 (based on SOC, MOC, MBC and POC). During training, observations were shuffled, and additional diagnostic metrics (α, β, Pearson correlation) were recorded for each configuration.

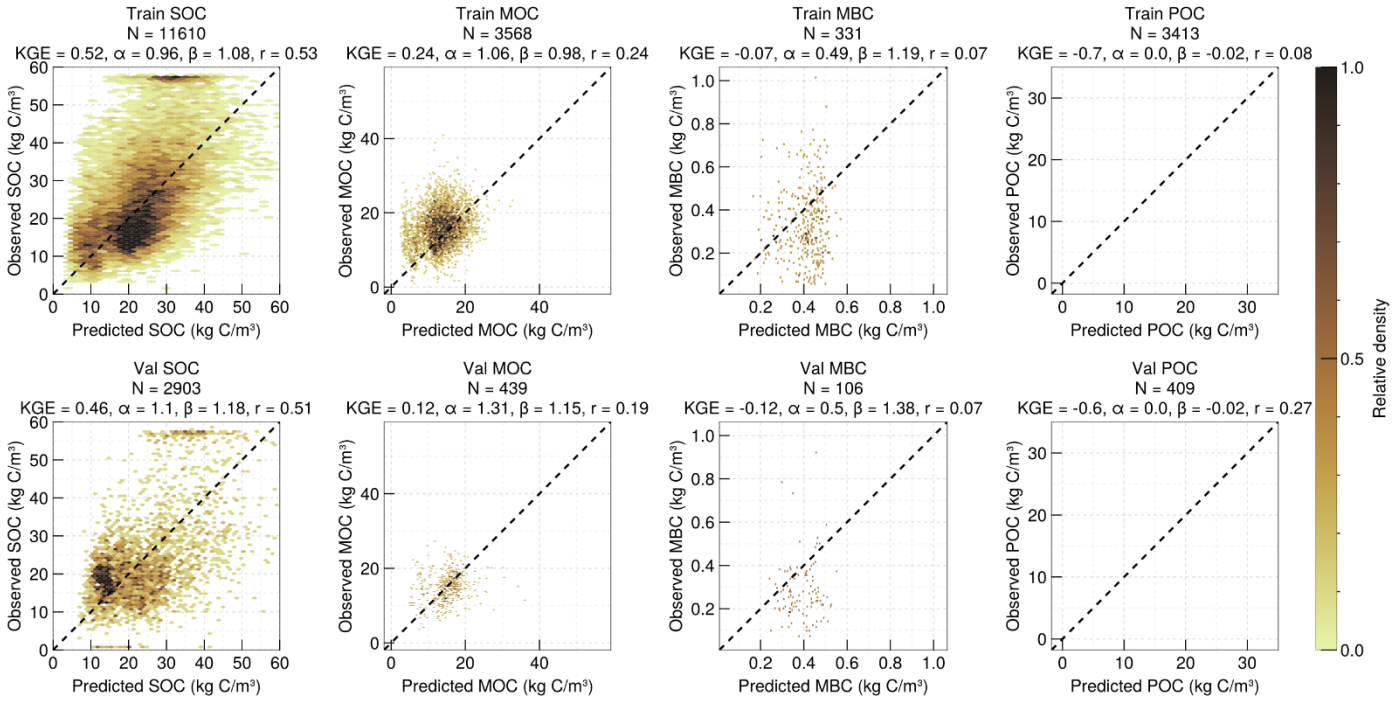

Figure 5. **All global:** all mechanistic parameters learned as a global parameter - performance of the hybrid model in training (upper) and validation (lower) for SOC, MOC, MBC and POC. KGE (Kling-Gupta efficiency) measures model performance as the Euclidean distance of three components - correlation r, variability ratio α, and bias ratio β - from the ideal point.

Unsurprisingly, the **All global** setup did not perform well (Figure 5). Although SOC was predicted quite well for validation profiles (KGE = 0.46), the model performed poorly for MOC (KGE = 0.12) and bad for MBC and POC with negative KGE values. Even in training POC could not be represented well – in Figure 5, one can barely make out a prediction near 0 for all sites. It is even worse than using the mean which according to Knoben, Freer, and Woods (2019) would have a KGE of -0.41. However, this only tells us that with this model structure and current Q_{10}, and given

the temperature and NPP forcing, we cannot represent spatial variability across the four data streams with otherwise globally estimated parameters.

In the next learning experiment, we made $\boldsymbol{CUE}$, $Q_{\max}$ and litter decomposition related parameters f, $V_{\max,0}$, $K_{M,0}$ spatially varying (Figure 6). For all these parameters spatial variation is plausible, and validation performances were much better with KGEs for SOC = 0.5, MOC = 0.27, MBC = 0.21, and POC = 0.3. These KGEs were generally quite close to the pure neural network KGEs of SOC = 0.6, MOC = 0.3, MBC = 0.25, and POC = 0.18.

We saw the same pattern of overfitting of the sparser datasets, especially MBC (see Figure 6). Based on this we can conclude that hybrid models can generally be as performant as a pure neural network model. However, based on this modelling setup there also does not seem to be a performance advantage of hybrid models over neural networks. Our random training and validation split may, however, not be the right setup to let hybrid models shine. The prescribed Q_{10} temperature sensitivity should become particularly advantageous if profiles are grouped into contrasting temperature regimes. Before using hybrid models to produce parameter maps or SOC, MOC, POC and MBC products in Work Package 5, we plan to test this explicitly in experiments stratified by temperature.

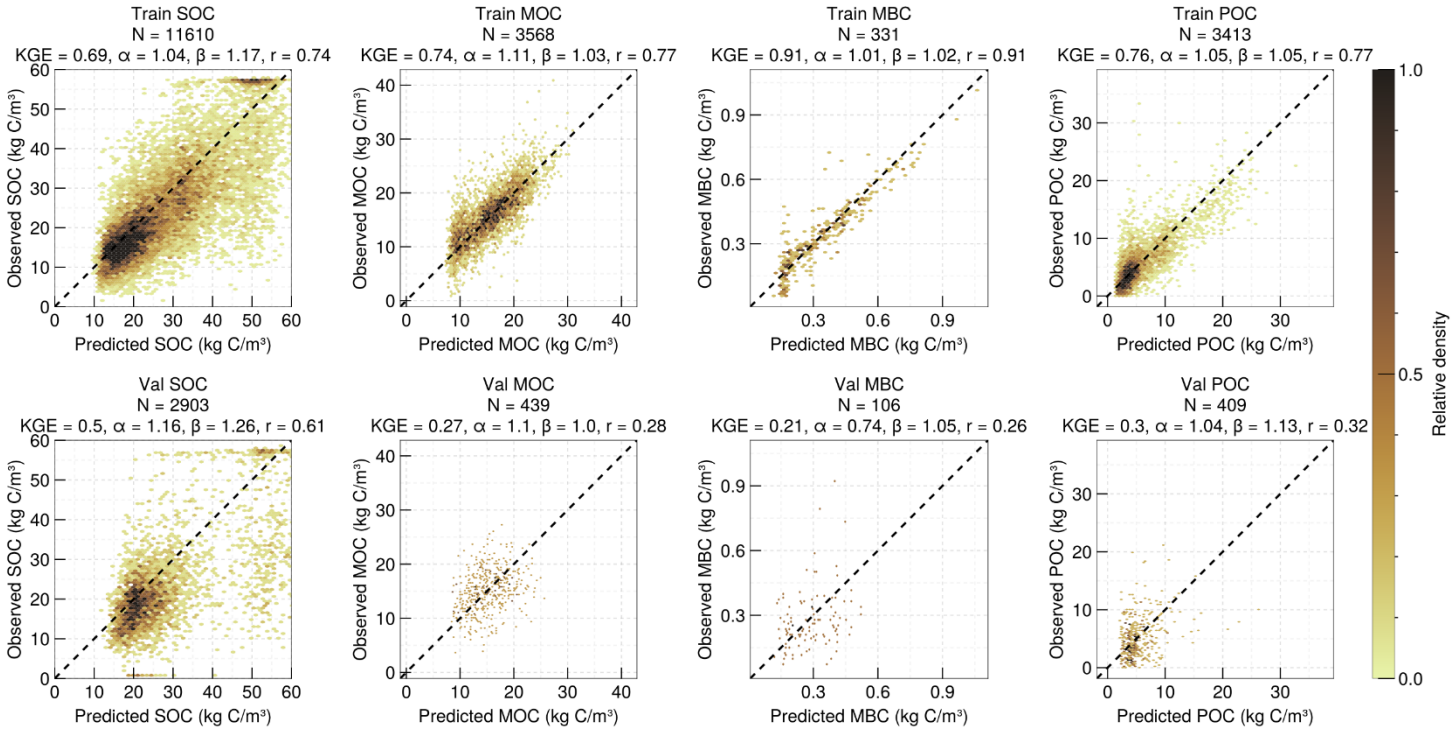

Figure 6. **CUE, $Q_{\max}$, f, $V_{\max,0}$, $K_{M,0}$ neural** - spatially varying parameters - performance of the hybrid model in training (upper) and validation (lower) for SOC, MOC, MBC and POC. KGE (Kling-Gupta efficiency) measures model performance as the Euclidean distance of three components - correlation r, variability ratio α, and bias ratio β - from the ideal point.

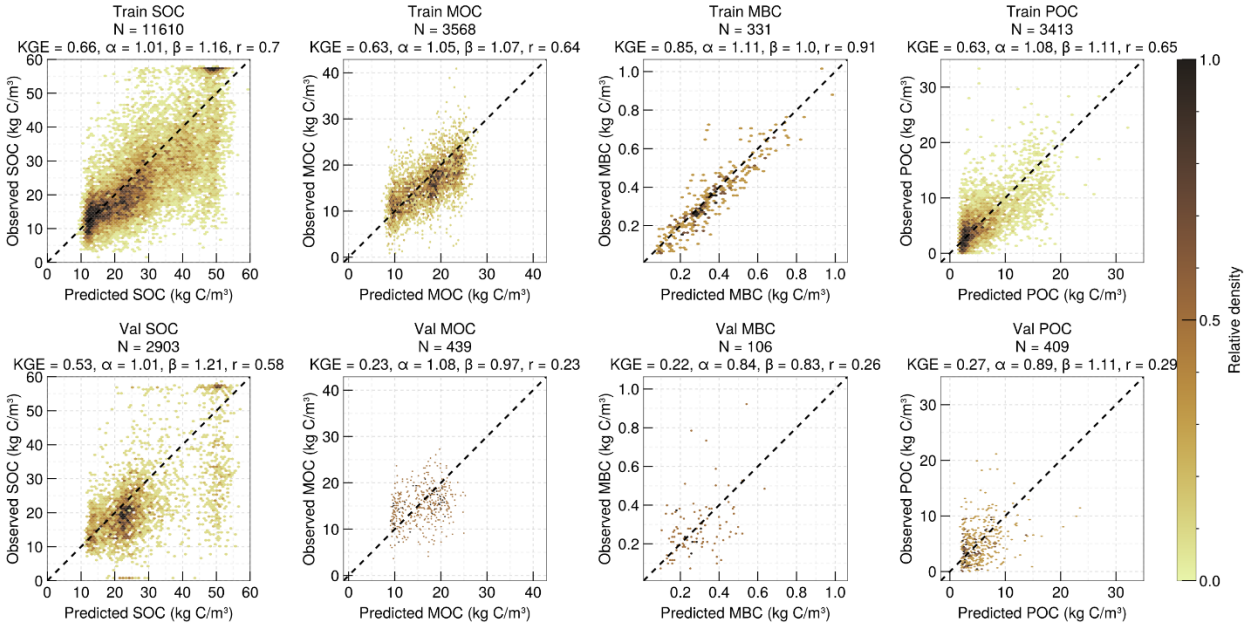

Figure 7. **All neural** - all parameters are learned with a neural network as spatially varying parameters - performance of the hybrid model in training (upper) and validation (lower) for SOC, MOC, MBC and POC. KGE (Kling-Gupta efficiency) measures model performance as the Euclidean distance of three components - correlation r, variability ratio α, and bias ratio β - from the ideal point.

Finally, we allowed all parameters to vary spatially to see if there may be further gains in explaining spatial patterns (**All neural**, Figure 7). The validation KGEs for **All neural** are SOC = 0.53, MOC = 0.23, MBC = 0.22, and POC = 0.27. Again overall, quite close to the pure neural network KGEs SOC = 0.60, MOC = 0.3, MBC = 0.25, and POC = 0.18. Hence, also the performance metrics of the **CUE,** $\boldsymbol{Q_{\max}}$**,** $\boldsymbol{f}$**,** $\boldsymbol{V_{\max,0}}$**,** $\boldsymbol{K_{M,0}}$ **neural** setup (with KGEs for SOC = 0.5, MOC = 0.27, MBC = 0.21, and POC = 0.3) is very close to the **All neural** setup. This indicates that the balance between spatial and global parameters could either be tuned as a hyperparameter or be informed by expert knowledge and literature.

As seen in the discussion between He et al. (2024) and Tao et al. (2023), it is also crucial for hybrid models to conduct further plausibility tests to make sure that the neural network part of the hybrid model did not produce parameters that give good results for the wrong reason.

One such test could be to see if the derived parameters are internally consistent with the model structure. We checked if the hybrid model recovers realistic saturation behaviour of mineral-associated organic carbon by performing *a post hoc* diagnosis of the fitted parameters and

predictions. For each site in the training data, we extracted the calibrated $Q_{\max}$ values from the outputs of the neural network and the corresponding predicted MOC. We then computed the ratio $\mathrm{MOC}/Q_{\max}$ and plot this in a histogram (Figure 8). It is evident that the calculated saturation ratios are unrealistic as they should never be larger than 1. This can have multiple reasons:

- The mechanistic model structure is wrong. This would be the ideal case for testing different model formulations of MOC formation. In the mechanistic model that we used only the DOC (C_D) pool adsorbs to mineral surface. There is, however, ample evidence that dead microbial biomass makes up the bulk of the MOC pool (Miltner, Bombach, Schmidt-Brücken, & Kästner, 2012). Some models have taken this into consideration (Ahrens et al., 2020; Wang, Huang, Zhou, Mayes, & Zhou, 2020)
- The neural network part of the hybrid model can overwhelm the mechanistic model due its data-adaptiveness. Since we use a steady-state formulation, the Langmuir sorption constraint is no longer enforced explicitly in the optimisation. As a result, the model does not automatically fail when MOC exceeds $Q_{\max}$, and unrealistic saturation ratios can occur. An introduction of further reality constraints such as MOC/Q_{max} has to be smaller than 1 or the DOC pool has to be small compared to the POC pool.

This example underlines that SOC remains contentious (Lehmann & Kleber, 2015), both because our mechanistic understanding is still incomplete and because SOC data are inherently sparse compared to, for example, eddy-covariance measurements used to study ecosystem carbon balance. However, with a diagnosis of unrealistic behaviour of a hybrid model, it should be clear that one then refrains from deeper interpretation of the learned parameters, which were right for the wrong reason. Then the cycle of testing other model formulations, maybe adapting model formulations starts again.

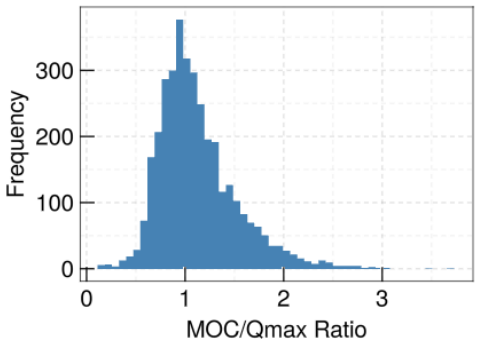

Figure 8. MOC to Q_{max} ratio. Q_{max} is theoretical mineral capacity. This ratio should not be larger than 1. This indicates that this mechanistic model is not able to reasonably represent all soil organic carbon pools.

5 Conclusion

While the SOC case study reveals limitations of the specific process-based SOC model used in this study, leading to mixed performance and even unrealistic behaviour, the hybrid framework itself, EasyHybrid.jl, works as intended. By combining several soil health indicators such as soil organic carbon and microbial biomass, we have illustrated that hybrid modelling can act as the 'glue' to bring several soil health indicators into one interacting and more holistic framework. The effects of soil microbial diversity on soil organic carbon formation and decomposition could be studied by using functional microbial groups as predictors for carbon use efficiency or decomposition rates. In the supplementary material, we present a case study on a structural soil health indicator—plant-available water capacity—and quantify how strongly soil organic carbon can influence this indicator. This also further illustrates the potential of EasyHybrid.jl under a simpler and better constrained process model and with abundant data (see Supplementary material).

Supplementary section: parameter learning for the influence of organic carbon on soil water retention curves

To test the hybrid framework in a setting with richer data and a simpler process description than SOC formation and decomposition, we applied it to soil water retention curves and derived the effect of soil organic carbon on plant-available water capacity. For this, we used the Lebeau–Konrad (LK) model as a mechanistic description of the soil water retention curve $\theta(h)$, where h is the matric potential and the parameters $\theta_s, h_m, \sigma, \theta_o, h_d$ control the capillary and adsorptive domains of soil water storage (Norouzi et al., 2022). Here, we show the actual code to illustrate how easy it is to train hybrid models in EasyHybrid.jl in three steps:

```
function mLK(;h, θ_s, h_m, θ_o, sigma, h_d)

    # capillary water content
    θ_c = @. 0.5 * θ_s * erfc(log(h/h_m)/(sqrt(2.0) * sigma))

    # adsorptive water content
    θ_a = @. θ_o * (1.0 - log(abs(h))/log(abs(h_d))) * (1.0 - θ_c/θ_s)

    # water content
    θ = θ_a + θ_c

    return (; θ, θ_a, θ_c, h_m, θ_o, sigma, θ_s)
end
```

Figure 9. Step 1 in EasyHybrid.jl: Define the mechanistic model. Shown is Konrad-Lebeau soil water retention curve model (mLK) that separates water contents θ into the adsorptive water content θ_a and the capillary water content θ_c. In the bracket behind mLK, the parameters and forcing of the model are given as keyword arguments. h is the matric potential, θ_s is the saturated volumetric water content, h_m is the matric potential that corresponds to the median capillary pore radius, sigma is the standard deviation of the log-transformed capillary pore radius distribution, and θ_o is a fitting parameter. erfc is the complementary error function. Besides water content θ_s all parameters and intermediate variables are returned.

In Step 2, we define the hybrid model, i.e. the structure of the neural network and the parameter roles: which parameters are learned as spatially varying functions of the predictors $\theta_{\text{NN}}(X)$, which

are learned as a single global value θ_{global}, and which are kept fixed θ_{fixed}.

```
hybrid_model_nn = constructHybridModel(
    [:clay, :silt, :sand, :oc],              # predictors
    [:h],                                    # forcing
    [:θ],                                    # targets
    mLK,                                     # mechanistic model
    parameters_LK,                           # parameter bounds
    [:θ_s, :h_m, :θ_o, :sigma],              # neural_param_names
    [],                                      # global_param_names
    hidden_layers = [32, 32],
    activation = tanh
)
```

Figure 10. Step 2 in EasyHybrid.jl: Define the hybrid model. Decide which predictors should be used – here clay, silt, sand, and soil organic carbon content. The measured matrix potential h is used as forcing. Soil water content θ is the variable that will be used as target/data to learn the parameters. We pass the name of the mechanistic model and upper and lower bounds for the parameters. Decide which parameters should be learned as a function of the predictors (neural parameter names) and which parameters should be learned as a global coefficient. The rest of the arguments describe the number of hidden layers and neurons that should be used. Finally, the activation function introduces non-linearity in neural networks.

In the last step 3, we train this hybrid model with function arguments that are very similar to classical machine learning:

```
tout = train(
    hybrid_model_nn,
    df;
    nepochs        = 1000,
    batchsize      = 1024,
    opt            = Adam(0.01)
);
```

Figure 11. Step 3 in EasyHybrid.jl: train the hybrid model. In the train function, one passes the name of the hybrid model and the dataset as a tabular dataframe. One then decides how many iterations/epochs should be used, the size of a batch of data that the optimizer Adam sees in one gradient calculation.

The training dataset consisted of measured water contents at several matric potentials per horizon, together with the corresponding soil texture, bulk density and soil organic carbon content. It comes from a collection of soil samples in Denmark and is described in detail in Norouzi et al. (2025). This yielded multiple pairs of soil matrix potential and soil water content per horizon. The hybrid model predicts soil water content in each horizon, and the high modelling efficiency/R^2 of the validation set shows that the parameters could be learned very well (Figure 12).

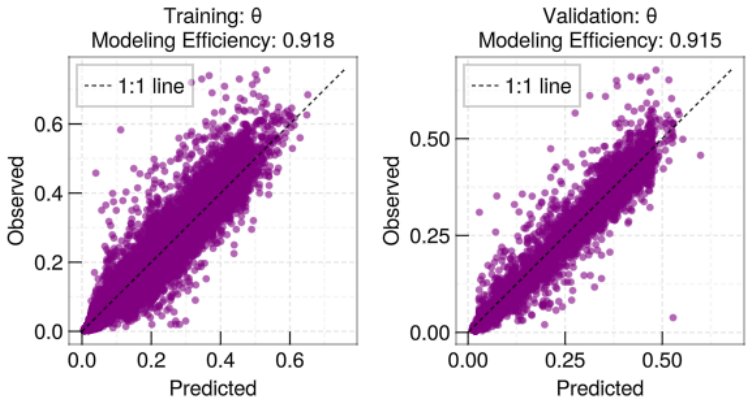

Figure 12. Model performance in training and validation in a random per-profile split into training and validation set. The performance is comparable to a non-parametric physics-informed neural network with an overall modelling efficiency of 0.92 for the same dataset (Norouzi et al., 2025).

After training, the hybrid model can be used to calculate secondary hydraulic indicators. Here we focus on plant-available water capacity (AWC), defined as the difference between volumetric water content at field capacity and at the permanent wilting point:

$$\mathrm{AWC} = \theta(\mathrm{pF} = 1.8) - \theta(\mathrm{pF} = 4.2).$$

This can be calculated by evaluating the trained hybrid model at two matric potentials (pF 1.8 and 4.2) for each soil and subtracting the predicted water contents from each other. Because the hybrid model is differentiable with respect to its parameters and has a smooth soil water potential, the resulting AWC is internally consistent with the underlying retention curve.

Finally, we used Shapley values to interpret the hybrid AWC predictions. Using the ShapML package, we created Shapley dependence plot for soil organic carbon for two training experiments. In one experiment, we used SOC, clay, silt and sand as predictors, in the other one we added bulk density. The latter shows an even slightly better validation performance with a modelling efficiency/R^2 of 0.934. Using only SOC, clay, silt and sand soil organic carbon shows a strong effect on changes in AWC, while when we add bulk density this effect of soil organic carbon on AWC is much reduced. Here, however, one has to keep in mind that bulk density and SOC are intimately linked. Increasing soil organic carbon can reduce bulk density and increase porosity (see Lebeau-Konrad model). This link has been studied with mechanistic models (Robinson et al., 2022) and could provide a useful connection between different soil health indicators when they are

combined in a mechanistic framework, with hybrid components acting as the glue that links the individual components together.

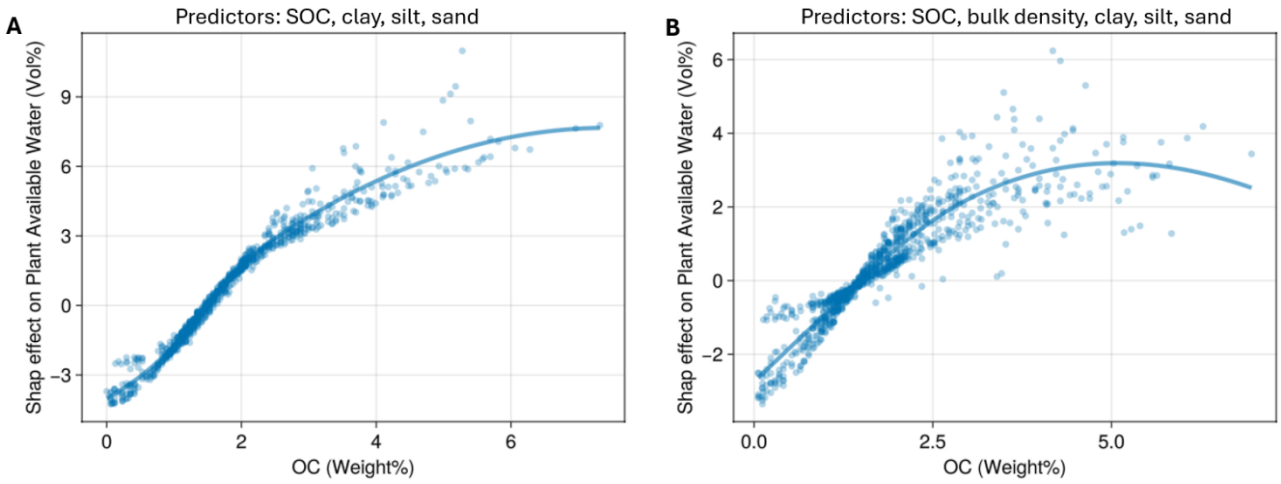

Figure 13. Shapley importance plots for the effect of soil organic carbon (OC Weight%) on plant available water (absolute change in volumetric water content in percentage.). **A** with only SOC, clay, silt and sand as predictors. **B** SOC, clay, silt, sand and additionally bulk density. Note the different y-scales.

References

Ahrens, B., Guggenberger, G., Rethemeyer, J., John, S., Marschner, B., Heinze, S., . . . Schrumpf, M. (2020). Combination of energy limitation and sorption capacity explains 14C depth gradients. *Soil Biology and Biochemistry, 148*, 107912. doi:10.1016/j.soilbio.2020.107912

Bar-On, Y. M., Li, X., O'Sullivan, M., Wigneron, J.-P., Sitch, S., Ciais, P., . . . Fischer, W. W. (2025). Recent gains in global terrestrial carbon stocks are mostly stored in nonliving pools. *Science, 387*(6740), 1291-1295. doi:10.1126/science.adk1637

Batjes, N. H., Ribeiro, E., & Van Oostrum, A. (2020). Standardised soil profile data to support global mapping and modelling (WoSIS snapshot 2019). *Earth System Science Data, 12*(1), 299-320. doi:10.5194/essd-12-299-2020

Beare, M. H., McNeill, S. J., Curtin, D., Parfitt, R. L., Jones, H. S., Dodd, M. B., & Sharp, J. (2014). Estimating the organic carbon stabilisation capacity and saturation deficit of soils: a New Zealand case study. *Biogeochemistry*, 1-17. doi:10.1007/s10533-014-9982-1

Breure, T. S., De Rosa, D., Panagos, P., Cotrufo, M. F., Jones, A., & Lugato, E. (2025). Revisiting the soil carbon saturation concept to inform a risk index in European agricultural soils. *Nature Communications, 16*(1). doi:10.1038/s41467-025-57355-y

Cotrufo, M. F., Ranalli, M. G., Haddix, M. L., Six, J., & Lugato, E. (2019). Soil carbon storage informed by particulate and mineral-associated organic matter. *Nature Geoscience, 12*(12), 989-994. doi:10.1038/s41561-019-0484-6

Feng, W., Plante, A. F., & Six, J. (2013). Improving estimates of maximal organic carbon stabilization by fine soil particles. *Biogeochemistry, 112*(1-3), 81-93.

Georgiou, K., Abramoff, R. Z., Harte, J., Riley, W. J., & Torn, M. S. (2017). Microbial community-level regulation explains soil carbon responses to long-term litter manipulations. *Nat Commun, 8*(1), 1223. doi:10.1038/s41467-017-01116-z

Georgiou, K., Jackson, R. B., Vindušková, O., Abramoff, R. Z., Ahlström, A., Feng, W., . . . Torn, M. S. (2022). Global stocks and capacity of mineral-associated soil organic carbon. *Nature Communications, 13*(1). doi:10.1038/s41467-022-31540-9

He, X., Abramoff, R. Z., Abs, E., Georgiou, K., Zhang, H., & Goll, D. S. (2024). Model uncertainty obscures major driver of soil carbon. *Nature, 627*(8002), E1-E3. doi:10.1038/s41586-023-06999-1

Innes, M., Edelman, A., Fischer, K., Rackauckas, C., Saba, E., Shah, V. B., & Tebbutt, W. (2019). A differentiable programming system to bridge machine learning and scientific computing. *CoRR, abs/1907.07587*.

Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and SemanticsUnknown article. doi:10.48550/arxiv.1705.07115

Knoben, W. J. M., Freer, J. E., & Woods, R. A. (2019). Technical note: Inherent benchmark or not? Comparing Nash–Sutcliffe and Kling–Gupta efficiency scores. *Hydrology and Earth System Sciences, 23*(10), 4323-4331. doi:10.5194/hess-23-4323-2019

Kraft, B., Jung, M., Körner, M., Koirala, S., & Reichstein, M. (2021). *Towards hybrid modeling of the global hydrological cycle*. Copernicus GmbH. Retrieved from https://dx.doi.org/10.5194/hess-2021-211

Lehmann, J., & Kleber, M. (2015). The contentious nature of soil organic matter. *Nature, 528*(7580), 60-68. doi:10.1038/nature16069

McBratney, A. B., Mendonça Santos, M. L., & Minasny, B. (2003). On digital soil mapping. *Geoderma, 117*(1-2), 3-52. doi:10.1016/s0016-7061(03)00223-4

Miltner, A., Bombach, P., Schmidt-Brücken, B., & Kästner, M. (2012). SOM genesis: microbial biomass as a significant source. *Biogeochemistry, 111*(1-3), 41-55. doi:10.1007/s10533-011-9658-z

Minasny, B., Bandai, T., Ghezzehei, T. A., Huang, Y.-C., Ma, Y., McBratney, A. B., . . . Widyastuti, M. (2024). Soil Science-Informed Machine Learning. *Geoderma, 452*, 117094. doi:10.1016/j.geoderma.2024.117094

Minasny, B., & McBratney, A. B. (2002). The Neuro-m Method for Fitting Neural Network Parametric Pedotransfer Functions. *Soil Science Society of America Journal, 66*(2), 352-361. doi:10.2136/sssaj2002.3520

Norouzi, S., Pesch, C., Arthur, E., Norgaard, T., Moldrup, P., Greve, M., . . . de Jonge, L. (2025). Physics-Informed Neural Networks for Estimating a Continuous Form of the Soil Water Retention Curve From Basic Soil Properties. *Water Resources Research, 61*. doi:10.1029/2024WR038149

Norouzi, S., Sadeghi, M., Tuller, M., Liaghat, A., Jones, S. B., & Ebrahimian, H. (2022). A novel physical-empirical model linking shortwave infrared reflectance and soil water retention. *Journal of Hydrology, 614*, 128653.

Orgiazzi, A., Ballabio, C., Panagos, P., Jones, A., & Fernández-Ugalde, O. (2018). LUCAS Soil, the largest expandable soil dataset for Europe: a review. *European Journal of Soil Science, 69*(1), 140-153. doi:10.1111/ejss.12499

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature, 566*(7743), 195-204. doi:10.1038/s41586-019-0912-1

Robinson, D. A., Thomas, A., Reinsch, S., Lebron, I., Feeney, C. J., Maskell, L. C., . . . Cosby, B. J. (2022). Analytical modelling of soil porosity and bulk density across the soil organic matter and land-use continuum. *Scientific Reports, 12*(1). doi:10.1038/s41598-022-11099-7

Smith, L. C., Orgiazzi, A., Eisenhauer, N., Cesarz, S., Lochner, A., Jones, A., . . . Guerra, C. A. (2021). Large-scale drivers of relationships between soil microbial properties and organic carbon across Europe. *Global Ecology and Biogeography, 30*(10), 2070-2083. doi:10.1111/geb.13371

Tao, F., Huang, Y., Hungate, B. A., Manzoni, S., Frey, S. D., Schmidt, M. W. I., . . . Luo, Y. (2023). Microbial carbon use efficiency promotes global soil carbon storage. *Nature, 618*(7967), 981-985. doi:10.1038/s41586-023-06042-3

Tian, X., Consoli, D., Witjes, M., Schneider, F., Parente, L., Şahin, M., . . . Hengl, T. (2025). Time series of Landsat-based bimonthly and annual spectral indices for continental Europe for 2000–2022. *Earth System Science Data, 17*(2), 741-772. doi:10.5194/essd-17-741-2025

Tian, X., De Bruin, S., Simoes, R., Isik, M. S., Minarik, R., Ho, Y.-F., . . . Hengl, T. (2025). Spatiotemporal prediction of soil organic carbon density in Europe (2000–2022) using earth observation and machine learning. *PeerJ, 13*, e19605. doi:10.7717/peerj.19605

Tsai, W.-P., Feng, D., Pan, M., Beck, H., Lawson, K., Yang, Y., . . . Shen, C. (2021). From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling. *Nature Communications, 12*(1). doi:10.1038/s41467-021-26107-z

Wang, G., Huang, W., Zhou, G., Mayes, M. A., & Zhou, J. (2020). Modeling the processes of soil moisture in regulating microbial and carbon-nitrogen cycling. *Journal of Hydrology, 585*, 124777. doi:10.1016/j.jhydrol.2020.124777

Wang, G., Post, W. M., Mayes, M. A., Frerichs, J. T., & Sindhu, J. (2012). Parameter estimation for models of ligninolytic and cellulolytic enzyme kinetics. *Soil Biology and Biochemistry, 48*(0), 28-38. doi:10.1016/j.soilbio.2012.01.011