https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

Hybrid Physics-AI and Neural ODE Approaches for Spatially Distributed Hydrological Modeling

Ngo Nghi Truyen Huynh[1], Pierre-André Garambois[1], François Colleoni[1], and Jérôme Monnier[2]

[1]INRAE, Aix-Marseille Université, RECOVER, 3275 Route Cézanne, 13182 Aix-en-Provence, France
[2]INSA, Institut de Mathématiques de Toulouse (IMT), Université de Toulouse, 31400 Toulouse, France

Correspondence: Ngo Nghi Truyen Huynh (ngo-nghi-truyen.huynh@inrae.fr) and Pierre-André Garambois (pierre-andre.garambois@inrae.fr)

Abstract. Empirical models are among the earliest hydrological models and have evolved from the unit hydrograph to deep learning models. Despite their success, purely data-driven methods often lack interpretability and are highly sensitive to data quality, limiting their generalizability in data-scarce regions or under changing environmental conditions. Conceptual models, traditionally relying on simplified representations of physical processes governed by conservation laws of mass, momentum, and energy, remain widely used in operational hydrology due to their explainability and practical applicability. However, these process-based models inherently face structural uncertainties and a lack of scale-relevant theories—challenges that emerging artificial intelligence (AI) techniques may help address. Moreover, high-resolution models are crucial for predicting extreme events characterized by strong variability and short duration, making spatially distributed hybrid modeling critical in the current context. We introduce a hybrid physics-AI approach that integrates neural ordinary differential equations (ODEs), solved by an implicit numerical scheme, into a spatialized, regionalizable, and differentiable process-based model. The hydrological module is built on a continuous state-space system and an integrated process-parameterization neural network. This hybrid system solves the ODEs governing reservoir dynamics, while embedding a neural network to refine internal water fluxes, all without relying on an analytical solution, instead computing the model states simultaneously. This work also presents an upgraded version of the `smash` platform following its initial release, featuring a more comprehensive evaluation of hybrid models at relatively fine resolutions of kilometric spatial and hourly temporal scales. The results show that hybrid approaches demonstrate consistently strong and stable performance in calibration and various validation scenarios. Additionally, the neural ODE structure exhibits a hybridization effect that modifies state dynamics and runoff flow, achieving more reliable streamflow simulations for flood modeling.

1 Introduction

With the explosion of big data and artificial intelligence (AI), research on innovative approaches that leverage the power of AI for flood forecasting and hydrological modeling has demonstrated significant efficiency and advantages in terms of accuracy and computational cost compared to traditional rainfall-runoff models (Sit et al., 2020). The use of AI in hydrology is motivated by the recognition that conceptual hydrological models often lack scale-relevant laws, which can potentially be addressed through data-driven modeling approaches (Reichstein et al., 2019). In general, there are two main approaches for integration of

https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

machine learning (ML) and deep learning (DL) techniques in hydrological modeling. The first approach involves using a full black-box model that replaces traditional rainfall-runoff models. These purely data-driven models have been applied successfully to hydrological prediction, achieving state-of-the-art performance in various applications using long short-term memory (LSTM) network (Kratzert et al., 2018; Feng et al., 2020; Cho and Kim, 2022) and their variants like LSTM-based Seq2Seq model (Xiang et al., 2020). A second approach is the hybrid method, which seamlessly integrates neural networks into process-based numerical models, adhering to the principle of “learning under physical constraints.” This strategy aims to harness the predictive power of AI while maintaining the physical consistency and interpretability of hydrological processes within the context of differentiable modeling (Shen et al., 2023). For example, recent advancements in hybrid modeling include leveraging neural networks for regionalizing differentiable hydrological models (Feng et al., 2022; Huynh et al., 2024), generating more plausible potential evapotranspiration estimates (Wang et al., 2024), and refining the internal fluxes of rainfall-runoff models (Huynh et al., 2025b). Additionally, merging process-based differential equations with ML can be highly advantageous. This has been recently demonstrated with physics-informed neural networks (PINNs) in Raissi et al. (2019), where the process-based model serves as a weak constraint in the training cost function and is well-suited to assimilate observations (e.g., He et al., 2020). This advantage is also evident in universal differential equations that incorporate a universal approximator (Chen et al., 2019; Rackauckas et al., 2021; Yin et al., 2021).

Despite the success of both pure and hybrid ML models, each approach faces several limitations and challenges. First, pure ML models are generally highly sensitive to large and high-quality training datasets, making them less reliable than process-based models in data-scarce regions (Shen, 2018; Reichstein et al., 2019; Beven, 2020; Sit et al., 2020). Unlike physics-based models, which rely on governing equations, ML models do not inherently impose physical constraints, resulting in reduced generalizability under extreme or unseen hydrological conditions (Beven, 2020). Hybrid modeling is a promising approach to address these limitations by objectively integrating ML/DL components into physically based models, rather than replacing them. However, hybrid approaches also face their own set of challenges. A primary limitation is maintaining physical consistency throughout the hybrid framework while leveraging the flexibility of data-driven approaches (Reichstein et al., 2019). Moreover, the integration of ML components with physical models may introduce complexities in model coupling and error propagation across different components (Frame et al., 2021). Recent research suggests that ensuring numerical stability and physical process consistency, particularly when embedding neural networks into differential equation-based hydrological models, may help address some of these key challenges (Rackauckas et al., 2021; Höge et al., 2022; Song et al., 2024). For instance, Höge et al. (2022) proposed an approach based on neural ODEs, where neural networks are implemented within an implicit numerical scheme. Neural ODEs provide a mathematically grounded approach to hybrid modeling that not only reduces numerical errors but also enhances the stability of hydrological simulations under varying conditions. Such hybrid approaches would be particularly promising if implemented within a variational data assimilation (VDA) framework for spatially distributed hydrological modeling (Jay-Allemand et al., 2020; Colleoni et al., 2022; Huynh et al., 2023). Indeed, high-resolution hydrological modeling accounts for spatially distributed information and fine temporal scales, both of which are crucial for predicting extreme events characterized by strong variability and short duration. However, current hybrid approaches are often employed for lumped models and hybrid modeling at high spatio-temporal resolution has received little attention. Although the gap is beginning to narrow with recent studies of hybrid modeling for fully distributed models such as Huynh et al. (2025b) for algebraic resolution of ODEs, the generalization of resolution approaches for spatialized neural ODEs remains unexplored, considering that previous work (Song et al., 2024) has implemented a discretized numerical method into hybrid lumped models.

This study proposes a hybrid physics-AI approach that integrates neural ODEs, solved using an implicit numerical scheme, into a differentiable, regionalizable, and spatially distributed hydrological model within a VDA framework. The hydrological model is based on GR4 operators (Perrin et al., 2003), with a continuous state-space structure (Santos et al., 2018) and an integrated process-parameterization neural network. The approach implements a mathematically rigorous method for computing the Jacobian matrices required by the implicit scheme when incorporating neural network components. While we initially focus on simpler neural network architectures to ensure numerical stability and physical interpretability, the hybrid numerical solver is designed to be compatible with PyTorch's automatic differentiation (Paszke et al., 2019) and could employ more complex architectures or deeper networks for process-parameterization. Additionally, the regionalization neural network from Huynh et al. (2024) is extended to explore alternative neural network architectures, such as convolutional neural networks (CNNs) in addition to multilayer perceptrons (MLPs), to improve the adaptability and scalability of parameter estimation. These developments aim to bridge the gap between ML flexibility and physical model interpretability, uncovering hydrological behaviors and scale-relevant theories inferred with AI techniques. It enables the addressing of several open research questions by providing a robust and powerful tool for enhancing flood modeling, mitigating structural uncertainty in modeling, optimizing data efficiency, and enabling more effective multi-scale information extraction through hybrid flux correction.

This paper also introduces `smash` v1.1, an upgraded version of the `smash` platform, following its initial release v1.0 in Colleoni et al. (2025). The new version includes various hybrid physics-AI hydrological solvers and provides a more comprehensive user guide, along with detailed mathematical descriptions of the implemented models (see smash 1.1.0 Release Notes[1] for details). As the core solver for the French flash flood forecasting system (Piotte et al., 2020), `smash` is positioned to improve real-world flood simulation and hydrological forecasting, facilitating the integration of AI-enhanced physics-based modeling into operational hydrology.

2 Method

This study employs three key components. First, we implement the continuous state-space GR4 structure presented in Santos et al. (2018) into `smash`, in addition to the classical GR4 model without an explicitly formulated water balance in Perrin et al. (2003). This state-space model solves the water balance differential equations continuously using numerical schemes, instead of splitting the equations to compute the solutions analytically. Second, the seamless regionalization method using neural networks, HDA-PR (Hybrid Data Assimilation and Parameter Regionalization), proposed in Huynh et al. (2024), enabling the estimation of conceptual hydrological parameters from physical descriptors, is extended to incorporate CNNs in addition to MLPs. Finally, the process-parameterization neural network previously used for the analytical resolution of a discrete state-space model in Huynh et al. (2025b)—which relied on an algebraic structure—is now directly embedded into the ODEs. These

---

[1] https://smash.recover.inrae.fr/release/1.1.0-notes.html

https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

ODEs are solved using the Newton-Raphson method within an implicit Euler scheme, which mitigates numerical errors that arise when using simple explicit schemes with sequential computations and split operators.

2.1 Hybrid continuous state-space structure for a regionalizable and gridded hydrological model

The hydrological model $\mathcal{M}$ is constructed by partially composing two operators: (i) a learnable, neural network-based operator ϕ, predicting parameters and fluxes corrections of (ii) a dynamic and spatially distributed rainfall-runoff model with simplified hydraulic routing, consisting in a set of chained differential equation and denoted as $\mathcal{M}_{rr-hy}$, resulting in Equation 1.

$$\mathcal{M} = \mathcal{M}_{rr-hy}\left(\,.\,;\,\phi(\,.\,)\right) \tag{1}$$

Let $\Omega \subset \mathbb{R}^2$ represent a two-dimensional spatial domain, with $x \in \Omega$ as the spatial coordinate and $t \in]0,T]$ as the physical time. The surface flow path is a 8-direction (D8) drainage network, $\mathcal{D}_\Omega$. The rainfall-runoff model $\mathcal{M}_{rr-hy}$ dynamically maps atmospheric forcings $\boldsymbol{\mathcal{I}}$ to surface discharge Q, internal states $\boldsymbol{h}$, and internal fluxes $\boldsymbol{q}$, as expressed in Equation 2.

$$\boldsymbol{U}(x,t) = [Q,\boldsymbol{h},\boldsymbol{q}](x,t) = \mathcal{M}_{rr-hy}\left(\mathcal{D}_\Omega,\boldsymbol{\mathcal{I}}(x,t);\boldsymbol{f}_q(x,t),[\boldsymbol{\theta},\boldsymbol{h}_0](x)\right) \tag{2}$$

with $\boldsymbol{U}(x,t)$ the modeled state-flux variables, $\boldsymbol{f}_q$ the vector of spatially distributed corrections applied to internal fluxes $\boldsymbol{q}$ (details will be explained later), $\boldsymbol{\theta}$ and $\boldsymbol{h}_0$ the spatially distributed parameters and initial states of the hydrological model. A neural network-based estimator ϕ, consisting of a pair of neural networks, takes as input (i) neutralized (by interception reservoir) atmospheric inputs $\boldsymbol{\mathcal{I}}_n = (P_n, E_n)(x,t)$, along with the model states at previous time step $\boldsymbol{h}(x,t-1)$, to correct spatio-temporal internal fluxes $\boldsymbol{q}(x,t)$ (process-parameterization pipeline) and (ii) physical descriptors $\boldsymbol{\mathcal{D}}(x)$ to estimate spatialized hydrological parameters $\boldsymbol{\theta}(x)$ (regionalization pipeline), as shown in Equation 3.

$$\phi : \begin{cases} \boldsymbol{f}_q(x,t) & = \phi_1\left(\boldsymbol{\mathcal{I}}_n(x,t),\boldsymbol{h}(x,t-1);\boldsymbol{\rho}_1\right) \\ \boldsymbol{\theta}(x) & = \phi_2\left(\boldsymbol{\mathcal{D}}(x);\boldsymbol{\rho}_2\right) \end{cases} \tag{3}$$

with $\boldsymbol{\rho} = (\boldsymbol{\rho}_1,\boldsymbol{\rho}_2)$ the vector of trainable parameters, invariant to the spatial coordinate x over Ω, of the neural network(s) ϕ. In this study, the neural networks considered are two MLPs, or an MLP for water flux corrections and a CNN for regional parameter estimation.

Now, the mathematical details of the hybrid hydrological model operators can be summarized as follows. For a given cell $x \in \Omega$ and time step $t > 0$, $P(x,t)$ and $E(x,t)$ represent the local precipitation and potential evapotranspiration. First, an interception reservoir with a capacity of c_i, automatically computed using the flux matching technique (Ficchì et al., 2019), allows for the computation of the neutralized rainfall P_n and neutralized evapotranspiration E_n.

Next, the model states $\boldsymbol{h}$ are computed by solving a neural ODE system, generally expressed as:

$$\frac{d\boldsymbol{h}}{dt} = f_{phys}\left(\,.\,,\boldsymbol{h},\phi(\,.\,,\boldsymbol{h};\boldsymbol{\rho})\right) \tag{4}$$

where f_{phys} is the set of physically-based equations. This neural ODE system employs the following two key components:

- The neural network ϕ takes the model states as part of its inputs, thus affecting the model dynamics and state gradient information. It is expected to learn the model behavior by leveraging memory effects through state updates.

- The set of physical equations f_{phys} integrates the network ϕ_1 as a complementary component, which refines the internal water fluxes that describe the state dynamics. This differs from partly replacing internal fluxes as $\frac{d\boldsymbol{h}}{dt} = f_{phys} \circ \phi(.,\boldsymbol{h};\boldsymbol{\rho})$, or fully replacing them as $\frac{d\boldsymbol{h}}{dt} = \phi(.,\boldsymbol{h};\boldsymbol{\rho})$ (e.g., Höge et al. (2022)). The approach allows the neural ODEs to preserve an original structure driven purely by physical equations, rather than directly relying on neural network outputs, and enables learning under stronger physical constraints.

In this study, we consider f_{phys} in Equation 4 to be a set of GR production and transfer operators. The GR production state h_p and transfer state h_t are computed by solving Equation 5. The numerical resolution of this neural ODE system, ensuring both accurate resolution and numerical differentiability with respect to model parameters for optimization, will be explained in Section 2.2.

$$\frac{d\boldsymbol{h}}{dt} = \begin{pmatrix} \frac{dh_p}{dt} \\ \frac{dh_t}{dt} \end{pmatrix} = \begin{pmatrix} \left(1 - \left(\frac{h_p}{c_p}\right)^{\alpha_1}\right) P_n(1+f_{q,1}) - \frac{h_p}{c_p}\left(2 - \frac{h_p}{c_p}\right) E_n(1+f_{q,2}) \\ 0.9\left(\frac{h_p}{c_p}\right)^{\alpha_1} P_n(1+f_{q,1}) + k_{exc}\left(\frac{h_t}{c_t}\right)^{\alpha_3}(1+f_{q,3}) - \frac{c_t}{\alpha_2-1}\left(\frac{h_t}{c_t}\right)^{\alpha_2}(1+f_{q,4}) \end{pmatrix} \tag{5}$$

where $\alpha_1 = 2$, $\alpha_2 = 5, \alpha_3 = 3.5$ are classical GR constants (cf. Perrin et al. (2003); Santos et al. (2018)); $f_{q,i=1..4}$ are the corrections applied to internal fluxes and predicted by the process-parametrerization neural network ϕ_1; c_p, c_t, and k_{exc} represent the capacities of the production and transfer reservoirs, and the exchange coefficient, which are the conceptual production parameters predicted by the regionalization neural network ϕ_2. Then, hydrological runoff flux (lateral discharge feeding the routing module) produced at the pixel scale is computed by the closure equation of the ODE system as follows:

$$Q_{lat} = 0.1\left(\frac{h_p}{c_p}\right)^{\alpha_1} P_n(1+f_{q,1}) + k_{exc}\left(\frac{h_t}{c_t}\right)^{\alpha_3}(1+f_{q,3}) + \frac{c_t}{\alpha_2 - 1}\left(\frac{h_t}{c_t}\right)^{\alpha_2}(1+f_{q,4}) \tag{6}$$

It is important to note that the values of $f_{q,i=1..4}$ are constrained within the range of -1 to 1 due to the TanH activation function in the output layer of ϕ_1. Consequently, the transformation functions applied to these internal flux corrections (e.g., $1+f_{q,1}$, $1+f_{q,2}$, etc.) preserve the structure of the original conceptual model when $\boldsymbol{f}_q \equiv \boldsymbol{0}$, as all transformations result in a value of 1 in that case. These terms were defined according to the specific fluxes being corrected and relevant mathematical constraints.

Finally, the routing module utilized here is based on a conceptual 1D kinematic wave model, which is numerically solved using a linearized implicit numerical scheme (Te Chow et al., 1988). Typically, the discharge routing problem is simplified to a 1D problem by adopting a “D8” drainage scheme $\mathcal{D}_\Omega(x)$, derived from processing a digital elevation model (DEM) of the terrain, with the assumption that a single pixel exhibits the largest drained area. The kinematic wave model is a partial differential equation (PDE) obtained by simplifying the 1D Saint-Venant equations, assuming the momentum is reduced to the flow friction slope, which equals the bottom slope. This is done by employing a conceptual parameterization for the momentum, $A = a_{kw}Q^{b_{kw}}$, where A represents the flow’s cross-sectional area, Q is the discharge, and a_{kw} and b_{kw} are parameters to be estimated by the regionalization neural network ϕ_2. This expression is inserted into the mass equation $\partial_x Q + \partial_t A = \lambda Q_{lat}$, with Q_{lat} being the lateral discharge (total runoff generated at a pixel from the GR operators described above), and λ representing the conversion factor. The result is a single-equation discharge propagation model:

$$\partial_x Q + a_{kw} b_{kw} Q^{(b_{kw}-1)} \partial_t Q = \lambda Q_{lat} \tag{7}$$

This kinematic wave model is numerically solved with a classical finite differences approach (cf. Te Chow et al. (1988)).

The following section will detail the numerical method used to solve the ODEs in 5, which includes the neural network ϕ_1 that depends on the ODE state $\boldsymbol{h}$, hence called a neural ODE.

2.2 Resolution of neural ODEs within an implicit Euler scheme

To solve Equation 5, we employ an implicit Euler scheme. For a small time step dt, by defining $\dot{\boldsymbol{h}} = \frac{d\boldsymbol{h}}{dt}$, we have:

$$\boldsymbol{h}(t+dt) = \boldsymbol{h}(t) + dt \cdot \dot{\boldsymbol{h}}\left(\boldsymbol{h}(t+dt)\right) \tag{8}$$

Now, define:

$$\boldsymbol{g}\left(\boldsymbol{h}(t+dt)\right) = \boldsymbol{h}(t+dt) - \boldsymbol{h}(t) - dt \cdot \dot{\boldsymbol{h}}\left(\boldsymbol{h}(t+dt)\right) \tag{9}$$

Approximating the sought state $\boldsymbol{h}(t+dt)$ thus reduces to numerically solving the equation:

$$\boldsymbol{g}(y) = y - c - dt \cdot \dot{\boldsymbol{h}}(y) = \boldsymbol{0} \tag{10}$$

where $y = \boldsymbol{h}(t+dt)$ and $c = \boldsymbol{h}(t)$. Then, the solution of Equation 10 is approximated using the Newton-Raphson method as follows:

$$\begin{cases} y_0 = c, \\ y_{n+1} = y_n + \Delta y, \quad \text{where } \Delta y \text{ is the solution of } \nabla \boldsymbol{g}(y_n) \cdot \Delta y + \boldsymbol{g}(y_n) = \boldsymbol{0} \end{cases} \tag{11}$$

The Jacobian matrix $\nabla \boldsymbol{g}$ is given by:

$$\nabla \boldsymbol{g} = \begin{pmatrix} 1 - dt\frac{\partial \dot{h_p}}{\partial h_p} & \frac{\partial h_p}{\partial h_t} - dt\frac{\partial \dot{h_p}}{\partial h_t} \\ \frac{\partial h_t}{\partial h_p} - dt\frac{\partial \dot{h_t}}{\partial h_p} & 1 - dt\frac{\partial \dot{h_t}}{\partial h_t} \end{pmatrix} \approx \begin{pmatrix} 1 - dt\frac{\partial \dot{h_p}}{\partial h_p} & -dt\frac{\partial \dot{h_p}}{\partial h_t} \\ -dt\frac{\partial \dot{h_t}}{\partial h_p} & 1 - dt\frac{\partial \dot{h_t}}{\partial h_t} \end{pmatrix} \tag{12}$$

This simplification holds because $\frac{\partial h_p}{\partial h_t} = 0$ since production does not depend on transfer. Additionally, we assume that h_t depends on h_p only through its time derivative rather than instantaneously. Thus, h_t evolves as an accumulated effect of h_p, meaning its dependence on h_p is indirect and primarily through its time derivative, justifying the approximation $\frac{\partial h_t}{\partial h_p} \approx 0$. The remaining terms, which are $\frac{\partial \dot{h_p}}{\partial h_p}$, $\frac{\partial \dot{h_p}}{\partial h_t}$, $\frac{\partial \dot{h_t}}{\partial h_p}$ and $\frac{\partial \dot{h_t}}{\partial h_t}$, can be derived analytically with distinct formulations depending on whether the process-parameterization neural network ϕ_1 is included, as follows.

First, we introduce a variable change: $\tilde{\boldsymbol{h}} = \left(\tilde{h_p}; \tilde{h_t}\right)^T = \left(\frac{h_p}{c_p}; \frac{h_t}{c_t}\right)^T$. Since the conceptual parameters c_p and c_t remain constant over time, their derivatives vanish, leading to: $\frac{d\tilde{h_p}}{dt} = \frac{1}{c_p}\frac{dh_p}{dt}$ and $\frac{d\tilde{h_t}}{dt} = \frac{1}{c_t}\frac{dh_t}{dt}$. Then, the Jacobian matrix $\nabla \boldsymbol{g}$ in Equation 12 can be computed as follows in two different cases:

- For classical continuous state-space structure (ODE): The process-parameterization neural network is absent, we set $\boldsymbol{f}_q \equiv \boldsymbol{0}$ in Equation 5. The resulting Jacobian components are:

$$
\begin{aligned}
\frac{\partial \dot{\tilde{h}}_p}{\partial \tilde{h}_p} &= -\alpha_1 P_n \tilde{h}_p^{\alpha_1 - 1} - 2E_n(1 - \tilde{h}_p)\\
\frac{\partial \dot{\tilde{h}}_p}{\partial \tilde{h}_t} &= 0\\
\frac{\partial \dot{\tilde{h}}_t}{\partial \tilde{h}_p} &= 0.9\alpha_1 P_n \tilde{h}_p^{\alpha_1 - 1}\\
\frac{\partial \dot{\tilde{h}}_t}{\partial \tilde{h}_t} &= \alpha_3 k_{exc} \tilde{h}_t^{\alpha_3 - 1} - \frac{\alpha_2}{\alpha_2 - 1} c_t \tilde{h}_t^{\alpha_2 - 1}
\end{aligned}
\tag{13}
$$

- For hybrid continuous state-space structure (neural ODE): When incorporating the process-parameterization neural network $\phi_1(.,\boldsymbol{h})$, which depends on the ODE state and predicts the model flux corrections $\boldsymbol{f}_q$, one obtains a set of so-called neural ODEs (Equation 5). The resulting Jacobian components become:

$$
\begin{aligned}
\frac{\partial \dot{\tilde{h}}_p}{\partial \tilde{h}_p} &= P_n \left(\left(1 - \tilde{h}_p^{\alpha_1}\right) \frac{\partial f_{q,1}}{\partial \tilde{h}_p} - \alpha_1 \tilde{h}_p^{\alpha_1 - 1} (1 + f_{q,1}) \right) - E_n \left(\tilde{h}_p \left(2 - \tilde{h}_p\right) \frac{\partial f_{q,2}}{\partial \tilde{h}_p} + 2\left(1 - \tilde{h}_p\right)(1 + f_{q,2}) \right)\\
\frac{\partial \dot{\tilde{h}}_p}{\partial \tilde{h}_t} &= P_n \left(1 - \tilde{h}_p^{\alpha_1}\right) \frac{\partial f_{q,1}}{\partial \tilde{h}_t} - E_n \left(2 - \tilde{h}_p\right) \tilde{h}_p \frac{\partial f_{q,2}}{\partial \tilde{h}_t}\\
\frac{\partial \dot{\tilde{h}}_t}{\partial \tilde{h}_p} &= 0.9 P_n \tilde{h}_p^{\alpha_1 - 1} \left(\alpha_1 (1 + f_{q,1}) + \tilde{h}_p \frac{\partial f_{q,1}}{\partial \tilde{h}_p} \right) + k_{exc} \tilde{h}_t^{\alpha_3} \frac{\partial f_{q,3}}{\partial \tilde{h}_p} - \frac{c_t \tilde{h}_t^{\alpha_2}}{\alpha_2 - 1} \frac{\partial f_{q,4}}{\partial \tilde{h}_p}\\
\frac{\partial \dot{\tilde{h}}_t}{\partial \tilde{h}_t} &= 0.9 P_n \tilde{h}_p^{\alpha_1} \frac{\partial f_{q,1}}{\partial \tilde{h}_t} + k_{exc} \tilde{h}_t^{\alpha_3 - 1} \left(\alpha_3 (1 + f_{q,3}) + \tilde{h}_t \frac{\partial f_{q,3}}{\partial \tilde{h}_t} \right) - \frac{c_t \tilde{h}_t^{\alpha_2 - 1}}{\alpha_2 - 1} \left(\alpha_2 (1 + f_{q,4}) + \tilde{h}_t \frac{\partial f_{q,4}}{\partial \tilde{h}_t} \right)
\end{aligned}
\tag{14}
$$

  where the neural network Jacobian $\frac{\partial \boldsymbol{f}_q}{\partial \boldsymbol{h}}$ is computed explicitly using the backpropagation method, which determines the gradient of its outputs with respect to its inputs. We note that all components of the Jacobian matrix, including those from the neural network, are computed directly from the analytical mathematical formulations, which significantly reduces the computational cost compared to calling a numerical adjoint during the resolution of the ODEs with an implicit scheme. Alternative methods, which do not directly rely on mathematical formulations but could be more computationally intensive, include using linear tangents or, more recently, the "discretize-then-optimize" adjoint method proposed in Song et al. (2024), which aims to reduce the time consumption associated with adjoint calls in a differentiable implicit numerical scheme.

2.3 Model training

Considering the observed and simulated discharge time series $\boldsymbol{Q}^* = (Q^*_{g=1..N_G})^T$ and $\boldsymbol{Q} = (Q_{g=1..N_G})^T$, where N_G is the number of gauges across the study domain Ω, the discrepancy between the model and multi-catchment observations is evaluated

https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

using a cost function J as follows:

$$J(\boldsymbol{Q}^*,\boldsymbol{Q}) = \sum_{g=1}^{N_G} w_g j(Q_g^*,Q_g) \tag{15}$$

Here, $\sum_{g=1}^{N_G} w_g = 1$ (with $w_g = 1/N_G$ in this study), and $j(Q_g^*,Q_g) = 1 - NSE(Q_g^*,Q_g)$ for each gauge, where NSE denotes the quadratic Nash-Sutcliffe efficiency. Thus, J is a convex and differentiable function that relies on the output $\boldsymbol{Q}$ of the forward model $\mathcal{M}$, and consequently depends on the conceptual parameters $\boldsymbol{\theta}$ and the flux corrections $\boldsymbol{f}_q$, and therefore on the parameters $\boldsymbol{\rho}$ of the ANNs (cf. Equation 3). The VDA optimization problem is defined as shown in Equation 16, which involves optimizing the weights and biases of the neural network(s) ϕ.

$$\hat{\boldsymbol{\rho}} = \arg\min_{\boldsymbol{\rho}} J\left(\boldsymbol{Q}^*, \mathcal{M}_{rr-hy}\left(\,.\,;\phi(\,.\,;\boldsymbol{\rho})\right)\right) \tag{16}$$

The neural network(s) are trained as follows:

- Pre-calibration of conceptual parameters: In the first step, we pre-train only the regionalization neural network ϕ_2 to find a spatially distributed first guess for the conceptual parameters. The weights of the process-parameterization neural network ϕ_1, which is embedded in the ODE solver, are initialized using a normal distribution centered at zero with a small variance. This ensures that, while the non-zero initialization allows for proper flow of gradients in the network, the outputs of ϕ_1 being close to zero result in minimal corrections, thus preserving the original hydrological model structure. In other words, ϕ_1 has very limited impact during this phase. The pre-training uses a relatively high learning rate of 0.004 for faster gradient descent over 40 iterations. In the case of lumped parameters (i.e., $\boldsymbol{\rho} = \bar{\boldsymbol{\theta}}$ and $\phi_2 \equiv \mathrm{id}_{\phi_2}$), we simply use a heuristic optimization algorithm to provide a spatially uniform first guess.

- Main training: In this phase, both neural networks are trained simultaneously (or ϕ_1 and the conceptual parameters in the case of lumped parameters). The training uses a smaller learning rate of 0.003 to ensure stability over 240 iterations. The gradients of ϕ_2 are computed using a chained gradient approach described in Huynh et al. (2024), while the gradients of ϕ_1, which is embedded in the differentiable Fortran code, are computed using the adjoint model.

It is worth noting that the embedded network ϕ_1 must be twice differentiable mathematically (once for solving the ODEs and once for the calibration process). While many activation functions in neural networks are not differentiable at zero (e.g., ReLU), stochastic optimization can often bypass this problem since the network generally produces non-zero (or non-close-to-zero) values. However, as we aim to preserve the original structure by producing outputs close to zero during pre-calibration, numerical errors can arise. To mitigate this issue, we use the SiLU (Sigmoid Linear Unit) activation function, which is twice differentiable everywhere and provides smooth gradients.

https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

3 Case study and results

3.1 Study area and experimental design

The models are run at a spatial resolution of 1 km and an hourly time step. The methods are evaluated using a national database covering Metropolitan France with multi-source data. The study area is the Aude River basin, located in southern France, the same as in Colleoni et al. (2025), but with a coarser spatial resolution of 1 km instead of 500 m. This area comprises 25 catchments, including 12 upstream gauges and 13 downstream gauges (Figure 1). The study period spans 9 years, from August 2014 to July 2023, divided into two sub-periods: P1 (2015-2019) and P2 (2019-2023). We use additional one-year warm-up periods (2014-2015 for P1 and 2018-2019 for P2) to calibrate or validate the models. Period P1 is used for calibration, while P2 is used for validation.

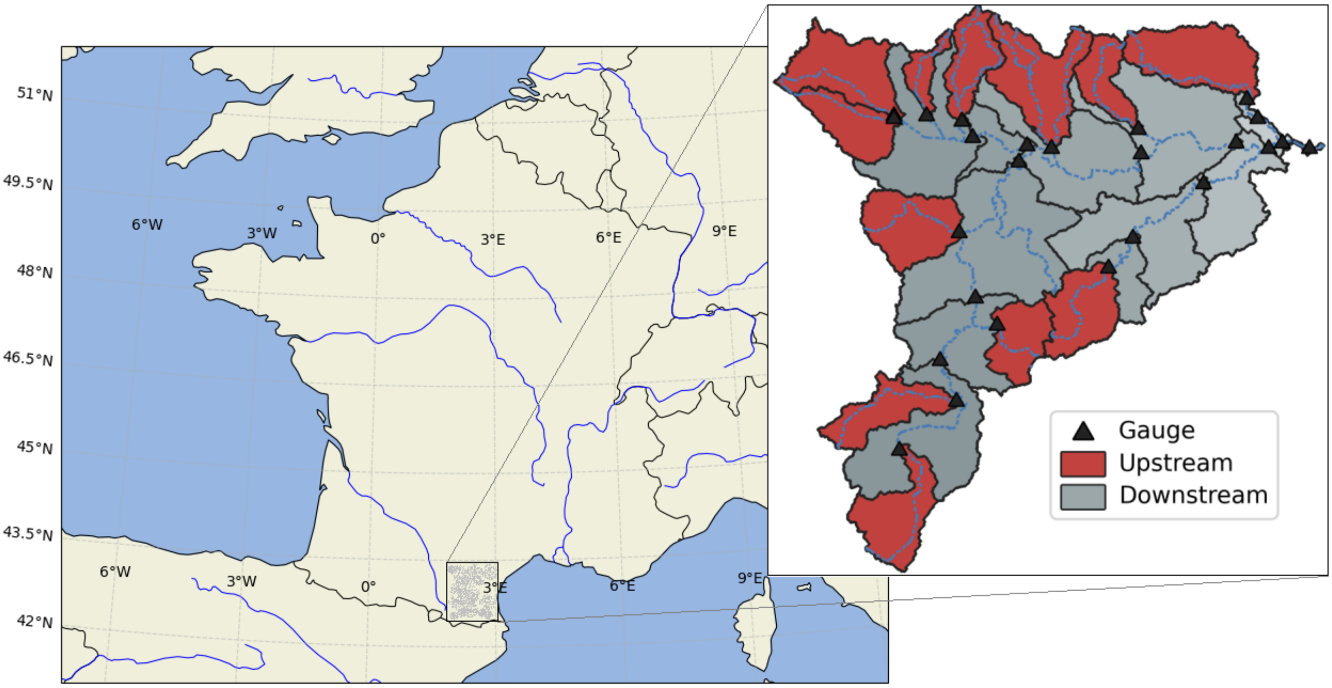

Figure 1. The Aude River basin, located in southern France, consists of 25 sub-catchments, including 12 upstream catchments (red-shaded regions) and 13 downstream catchments (gray-shaded regions).

We perform calibration across multiple catchments using upstream gauges and evaluate regionalization performance by transferring parameters to downstream gauges, which represents a more challenging regionalization case compared to transferring parameters from downstream to upstream gauges. A set of seven descriptors with a spatial resolution of 0.01° in the WGS 84 projection, similar to Huynh et al. (2024), is used as inputs for the regionalization mapping ϕ_2 (Figure 2).

https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

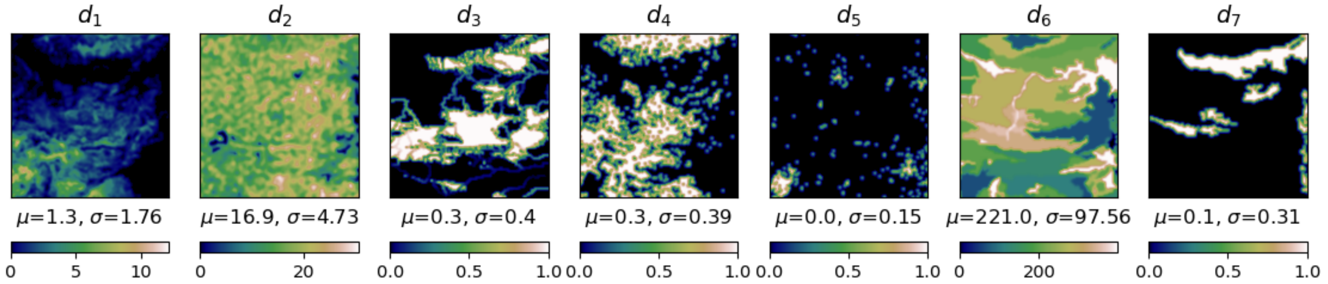

Figure 2. Maps of seven physical descriptors in the Aude River basin, where μ and σ represent the spatial average and standard deviation for each descriptor. d_1: the local slope (in degrees); d_2: the drainage density; d_3: the percentage of basin area in karst zone; d_4: the forest cover rate; d_5: the urban cover rate; d_6: the potential available water reserve (in mm); and d_7: the high storage capacity basin rate. Before the optimization process, all descriptors are standardized between 0 and 1 using min-max scaling.

We evaluate the models based on two criteria: model structure flexibility and regionalization ability. Model structure flexibility is assessed by the model's capacity to produce interpretable water flux dynamics, both with and without the process-parameterization network ϕ_1. Regionalization ability is evaluated based on the capability of the neural networks ϕ_2, which can be either an MLP or a CNN, to regionalize the conceptual parameters using physical descriptors. Table 1 summarizes the evaluated models and indicates the version of `smash` in which each was first introduced.

Table 1. Summary of evaluated models and their corresponding `smash` version. The notation consists of two parts, separated by a dot. The first part describes the model structure: GR denotes the GR4 structure (Perrin et al., 2003), which uses an algebraic formulation for state updates; ODE denotes the continuous GR4-like state-space model (Santos et al., 2018); and NODE denotes the neural ODE structure that incorporates a neural network into the ODEs. The second part indicates the mapping used to constrain the conceptual parameters: U represents lumped parameters without constraints; while MLP or CNN represent regionalization mappings based on MLP or CNN to map physical descriptors to spatially distributed conceptual parameters. All models include a kinematic wave routing over a flow direction grid.

Notation	Model	Version	Description
GR.U	Algebraic (lumped parameters)	v1.0	GR4 model with lumped parameters
GR.MLP	Algebraic + MLP	v1.0	Grid-based GR4 model using an MLP for regionalization
GR.CNN	Algebraic + CNN	v1.1	Grid-based GR4 model using a CNN for regionalization
ODE.U	ODE (lumped parameters)	v1.1	Continuous state space GR4-like model with lumped parameters
ODE.MLP	ODE + MLP	v1.1	Continuous state-space GR4-like model with MLP-based regionalization
ODE.CNN	ODE + CNN	v1.1	Continuous state space GR4-like model with CNN-based regionalization
NODE.U	Neural ODE (lumped parameters)	v1.1	Neural ODE-integrated GR4-like model with lumped parameters
NODE.MLP	Neural ODE + MLP	v1.1	Neural ODE-integrated GR4-like model with MLP-based regionalization
NODE.CNN	Neural ODE + CNN	v1.1	Neural ODE-integrated GR4-like model with CNN-based regionalization

3.2 Validation of the ODE solver

A major advantage of numerically solving ODEs, as opposed to algebraic resolution methods, lies in their generalizability. While an analytical solution derived through an algebraic approach is exact, it can only be obtained under specific assumptions about the ODE system (e.g., certain fixed values of the coefficients $(\alpha_i)_{i=1..3}$ in Equation 5). In contrast, numerical methods can solve ODEs without requiring such assumptions, albeit with approximate solutions. It is thus necessary to validate the ODE solver (ODE solutions obtained by numerical scheme) against the algebraic approach before performing any numerical experiments based on this solver. This is particularly important as this work presents the first implementation of both classical and hybrid ODE solvers into a fully distributed hydrological modeling and VDA framework.

To carry out this validation, we compare the GR algebraic model with lumped parameters (GR.U), which solves an analytical solution of the time-integrated ODEs, and the continuous state-space model also with lumped parameters (ODE.U), which solves the ODEs using an implicit numerical scheme. For both models, we set identical conceptual parameters and initial states, which are non-calibrated. Figure 3 shows similar simulated hydrological responses for GR.U and ODE.U obtained at the most downstream gauge. Although slight differences are found in the production state (often higher state for ODE.U) at certain

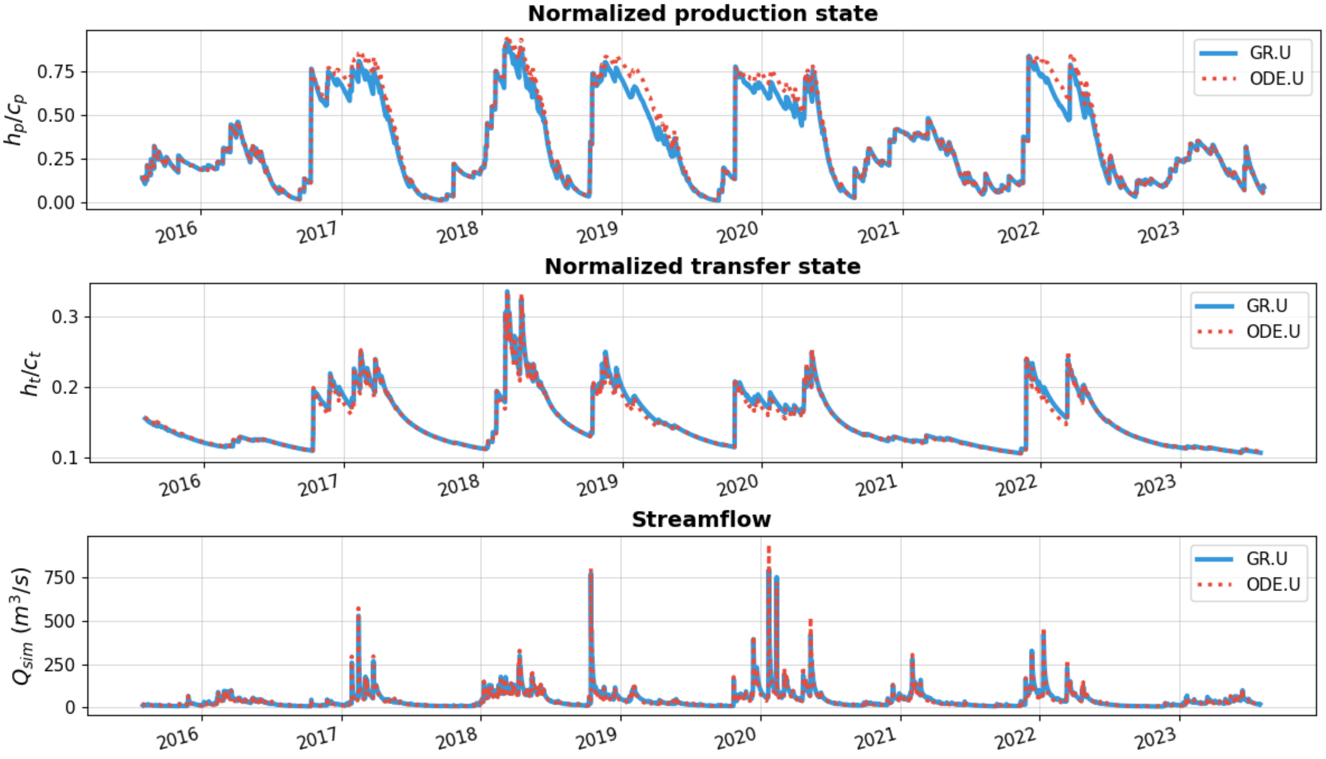

Figure 3. Comparison of normalized production state (top), normalized transfer state (middle), and simulated streamflow (bottom) at the most downstream gauge, obtained by algebraic resolution (GR.U) and numerical resolution (ODE.U) using the same lumped conceptual parameters and initial states.

periods, the transfer state for both models remains nearly identical, while streamflow simulation shows minor differences at several peak flows.

In addition to similar temporal hydrological responses, Figure 4 demonstrates nearly identical spatial patterns and values of the final model states obtained for both methods. The bias map between ODE.U and GR.U for the production state shows small deviations centered around zero, while the bias for the transfer state is nearly negligible, aligning with the earlier temporal observations. These minor discrepancies between the two approaches can be attributed primarily to the fundamental difference in their solution methodology—namely, the simultaneous numerical resolution of the ODEs for both state variables (h_p and h_t) versus the exact analytical solution obtained through sequential algebraic resolution—and the convergence criteria of the Newton-Raphson method.

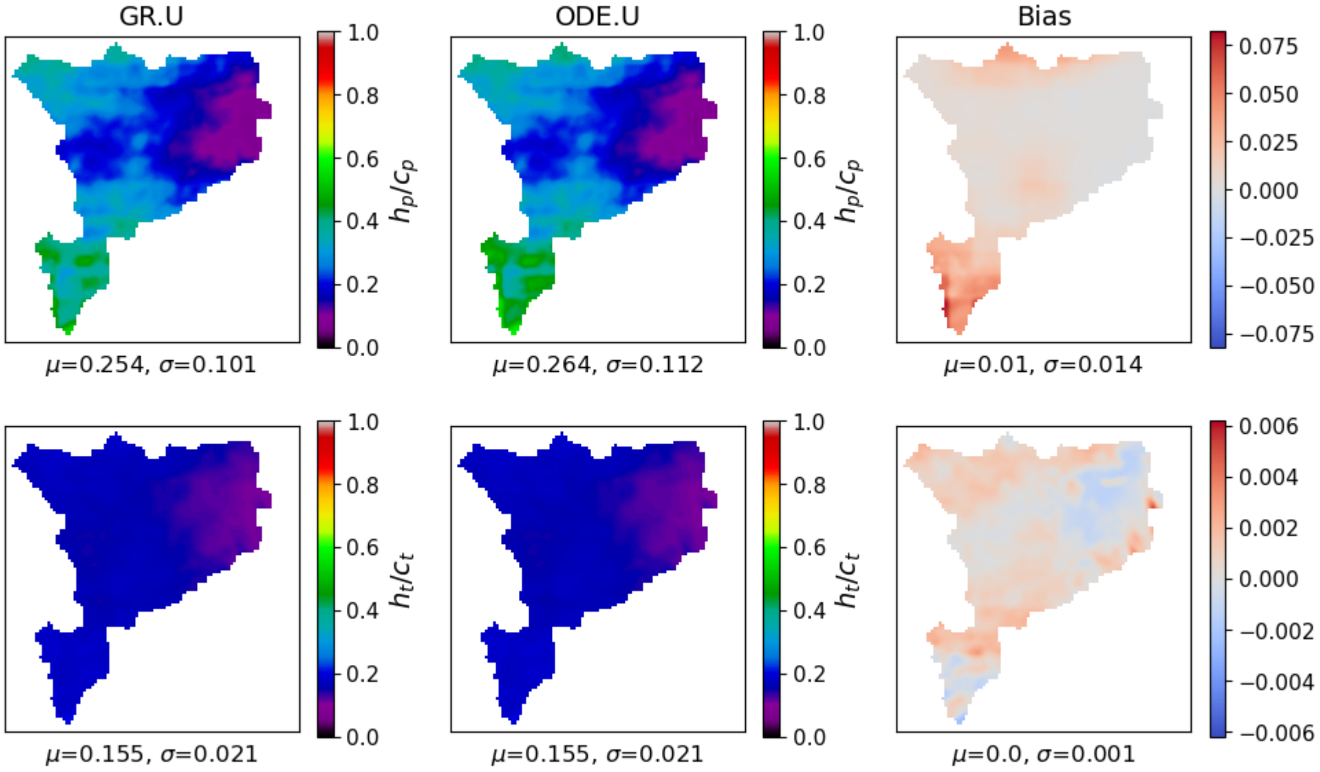

Figure 4. Maps of the normalized production state (h_p/c_p, top) and transfer state (h_t/c_t, bottom) at the final time step, as simulated by GR.U (left), ODE.U (middle), and the corresponding bias (right) computed as the difference between the ODE.U and GR.U states. For each map, μ and σ denote the spatial average and standard deviation.

While both methods initially produce similar hydrological responses, it will be particularly interesting to observe in the next section how the model dynamics evolve during calibration, and whether differences in the numerical formulation influence model behavior and performance.

3.3 Model performance and interpretation

Figure 5 provides a global view of model performance across calibration and various validation scenarios. Results demonstrate that model scores in calibration increase with the complexity of regionalization mappings (from uniform to MLP and CNN). For each regionalization approach, we observe improved scores progressing from classical GR structure to ODE and NODE structures, exemplified by a median NSE over 0.85 for NODE.CNN compared to 0.82 for ODE.CNN and 0.8 for GR.CNN.

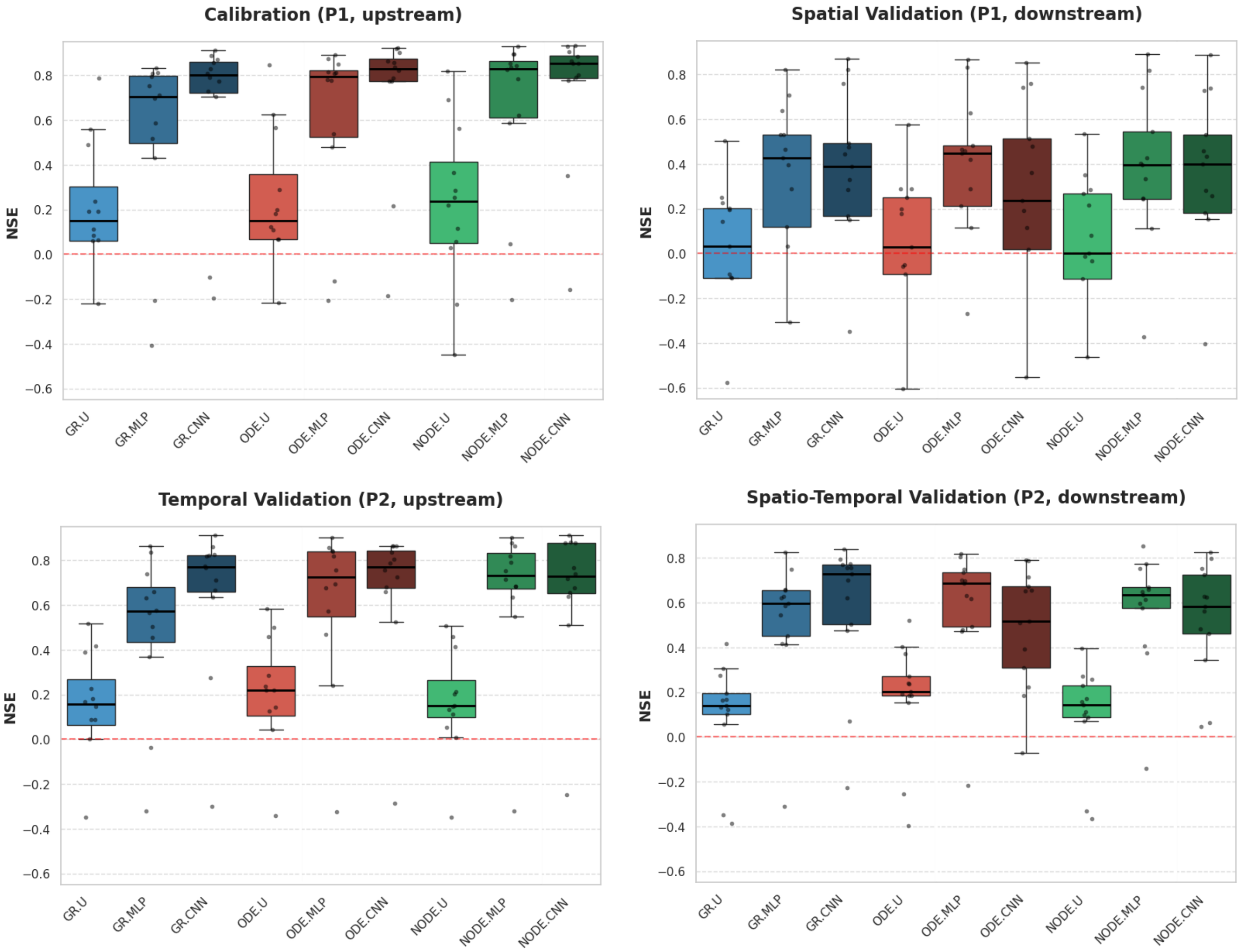

Figure 5. Comparison of model performance for different methods. The NSE scores are computed to evaluate: (i) calibration performance (scores computed over P1 for the 12 upstream gauges), (ii) spatial validation (scores computed over P1 for 13 downstream gauges), (iii) temporal validation (scores computed over P2 for 12 upstream gauges), and (iv) spatio-temporal validation (scores computed over P2 for downstream gauges).

https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

Moreover, models with ODE and NODE structures show improved interquartile ranges and whiskers when using CNN or MLP regionalization compared to GR, while NODE.U exhibits larger variance despite maintaining higher median scores than GR.U.

For temporal validation, ODE.MLP and NODE.MLP yield similar high median scores of 0.71 compared to 0.59 for GR.MLP. Additionally, NODE.MLP demonstrates a superior interquartile range, with a 0.25-quantile of 0.65 compared to 0.57 for ODE.MLP and 0.42 for GR.MLP. Models with the same regionalization approaches (uniform or CNN) show similar performance across different structures (GR, ODE, NODE), with only slight differences in median and distribution.

Spatial validation shows comparable performance across structures for each mapping, except for ODE.CNN which performs relatively worse than GR.CNN and NODE.CNN. While all models struggle with spatial validation, spatio-temporal validation yields generally good performance overall. Although ODE.CNN maintains lower performance compared to GR.CNN and NODE.CNN, the MLP-based models exhibit a more stable performance across validation scenarios. For instance, ODE.MLP achieves the best median scores among the three MLP-based models for spatio-temporal validation, while NODE.MLP yields the best interquartile range. Interestingly, GR.CNN exhibits promising performance in spatio-temporal validation with a median score of 0.71.

Overall, GR.CNN and NODE.MLP demonstrate consistently strong and stable performance across all calibration and validation scenarios. While GR.CNN generally outperforms its MLP counterpart in validation scenarios, NODE.MLP shows stable performance with good scores and favorable interquartile ranges across all validation scenarios.

To illustrate the models' flood simulation capabilities, Figure 6 presents two representative flood events during the validation period, as observed at an upstream and a downstream gauge. At both locations, all models using uniform mapping significantly underestimate the flood magnitudes, resulting in the poorest simulation performance. This result highlights the importance of regionalization approaches in accurately simulating floods, particularly in ungauged basins, where traditional methods that use lumped parameters often fail to capture real hydrological dynamics.

For the upstream gauge example, regionalization approaches using MLP and CNN produce relatively similar discharge simulations for each model structure (GR, ODE, NODE). The ODE and NODE structures with MLP show slightly better performance, with RMSE values of 4.27 and 4.06, compared to 6.09 for GR.MLP. Both CNN and MLP mappings applied to the GR structure accurately simulate the peak flow, whereas the regionalization methods for the ODE and NODE structures tend to slightly overestimate peak flow, particularly in the case of ODE.CNN. Despite this, the timing of flood events remains accurate across all regionalization methods, with the rising limbs of the simulated hydrographs closely aligning with observations. Note that in this upstream catchment characterized by a quick hydrological response, the start of the flood event occurs nearly at the same time as the moment of heavy rainfall.

In contrast, the downstream gauge event shows an approximately 8-hour delay between heavy rainfall and flood response. Most models struggle to predict the lag time correctly. This emphasizes the need to improve model realism by accounting for rainfall intensity and studying its impact in triggering non-linear flash flood responses, improving river network hydraulics, which presents a promising avenue for future research. Returning to flood simulation at the downstream, while the majority of models yield poor performance in predicting the event's beginning, generating flood responses earlier than observed, GR.CNN

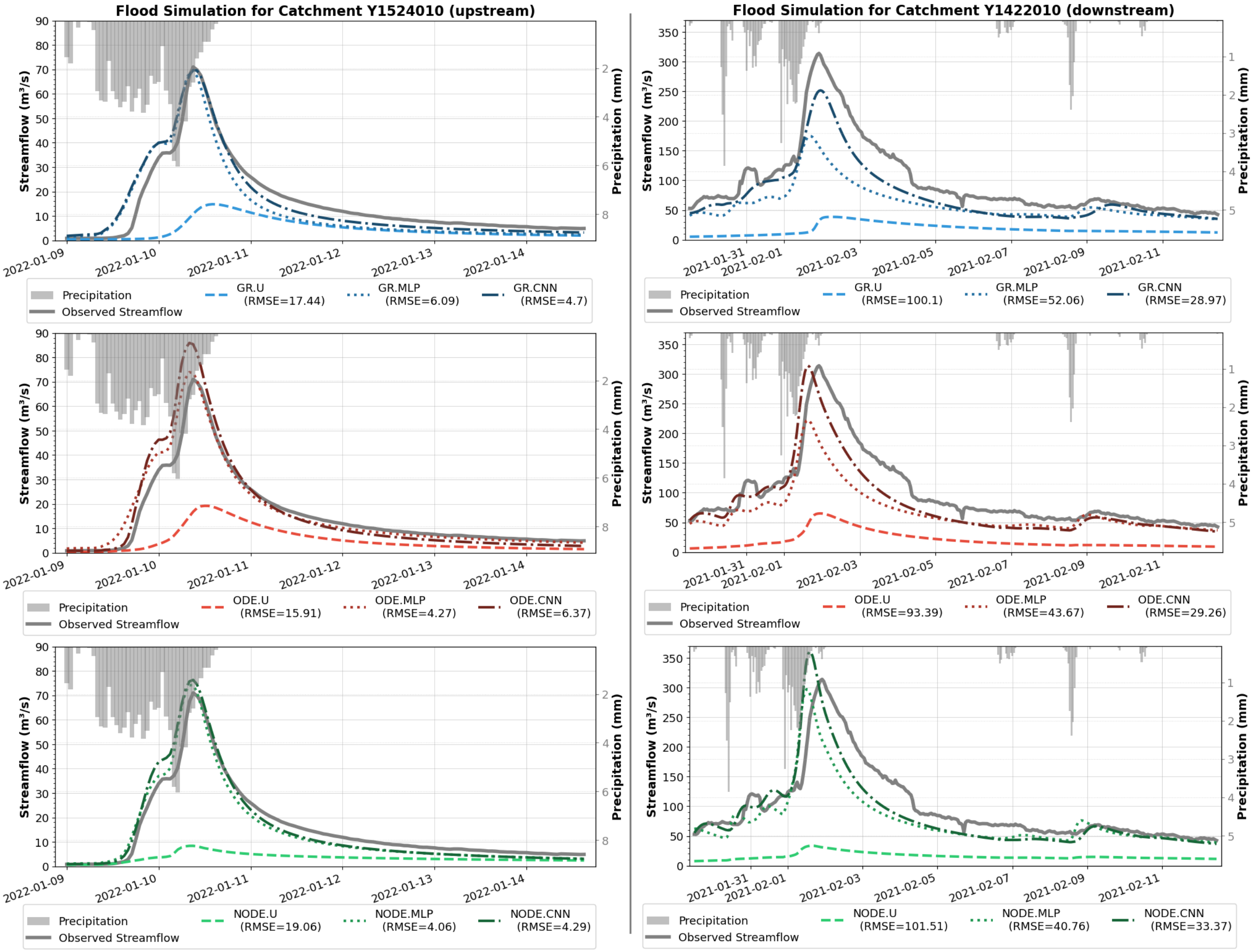

Figure 6. Comparison of flood simulation performance across different methods for an upstream gauge (left) and a downstream gauge (right).

demonstrates impressive timing accuracy with simulations very close to observations. However, GR structure models, including GR.CNN, still underestimate flood magnitude, whereas ODE.CNN, NODE.MLP, and NODE.CNN perform more accurately.

Figure 7 shows the maps of spatially distributed conceptual parameters calibrated using CNN and MLP regionalization approaches. Overall, CNN-based models produce smoother parameter maps compared to MLP-based models, as evidenced in the maps of c_p and c_t for GR.CNN (compared to GR.MLP), k_{exc} for all CNN-based structures, a_{kw} for all CNN-based structures, and b_{kw} for GR.CNN. This smoothing effect results from the convolution operations applied to physical descriptor maps. Furthermore, we observe spatial patterns from physical descriptors (Figure 2) reflected in the parameter maps, such as the pattern at the top of the forest cover map (d_4) or the pattern at the bottom left of the slope (d_1). For each state-space structure, different regionalization mappings produce distinct parameter distributions (e.g., different patterns of GR.MLP com-

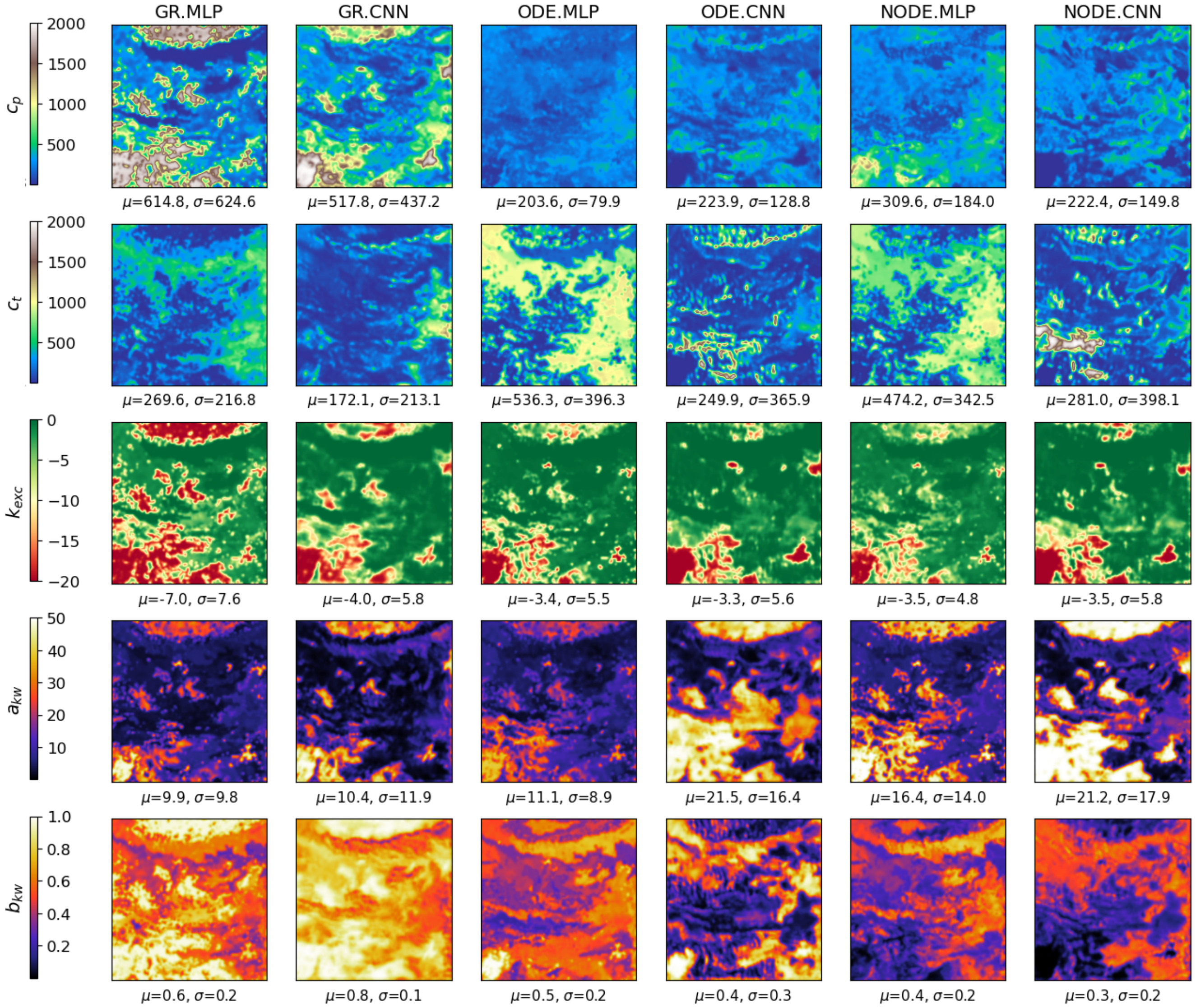

Figure 7. Estimated conceptual parameters across different regionalization methods. The calibrated parameters are $\boldsymbol{\theta}(x) = (c_p(x), c_t(x), k_{exc}(x), a_{kw}(x), b_{kw}(x))^T$ with μ and σ denoting their spatial average and standard deviation.

pared to GR.CNN, ODE.MLP compared to ODE.CNN, and NODE.MLP compared to NODE.CNN). Additionally, notable differences in parameter patterns emerge across model structures, with major differences for GR-based models, while ODE and NODE structures yield relatively similar parameter maps (ODE.MLP compared to NODE.MLP, and ODE.CNN compared to NODE.CNN). This pattern divergence is expected since classical GR models have different state dynamics compared to continuous state-space models, even when employing the same regionalization mapping.

https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

Figure 8 examines model dynamics through the spatial average of normalized states (production state h_p and transfer state h_t). The two continuous state-space models (ODE.MLP and NODE.MLP) demonstrate similar behavior in production state,

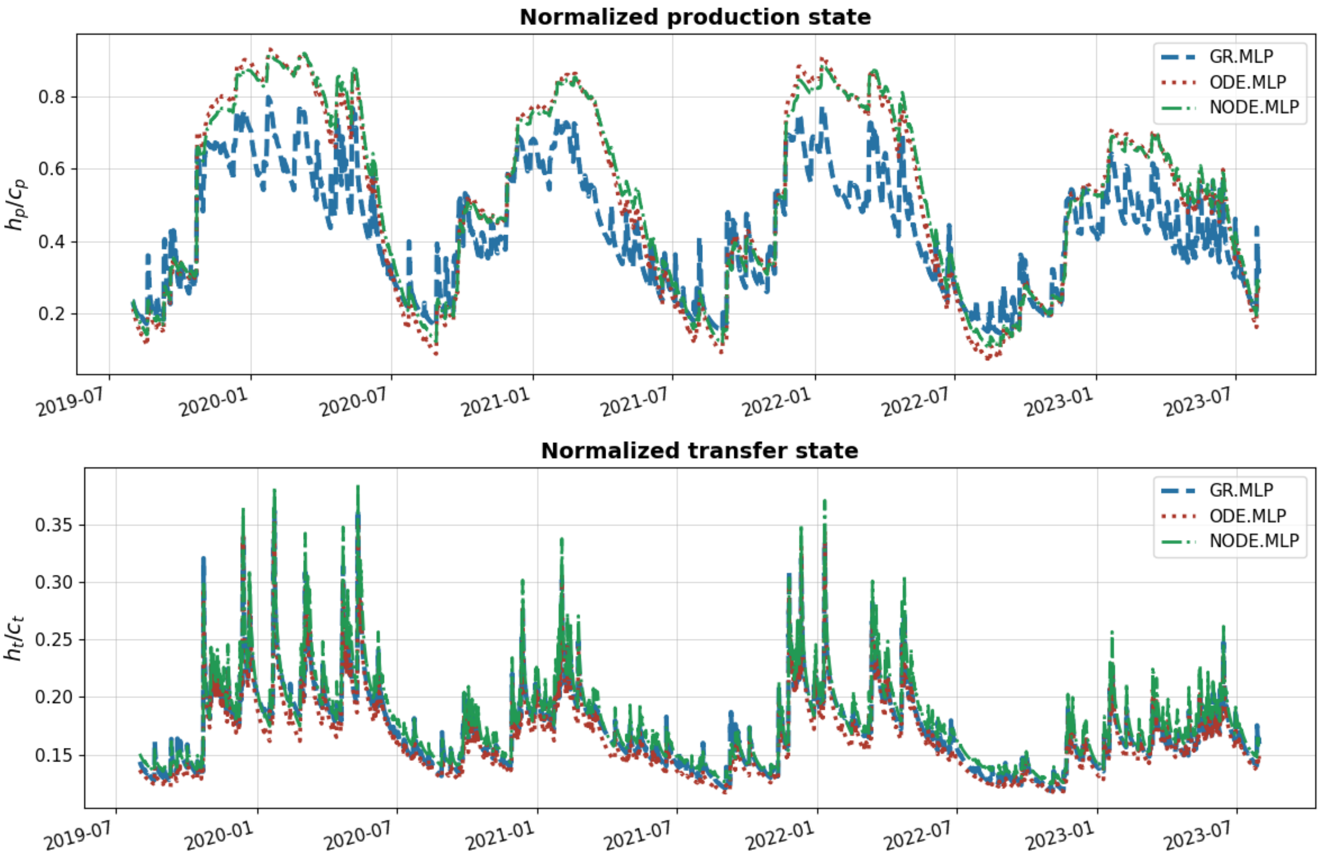

Figure 8. Comparison of the spatial average of the normalized production state $\overline{h_p(t)/c_p(t)}^x$ and transfer state $\overline{h_t(t)/c_t(t)}^x$ during the validation period P2, across the three model structures (GR, ODE, NODE), each employing an MLP-based regionalization mapping.

showing higher values and lower variability compared to GR.MLP. However, for the transfer state, the hybrid NODE.MLP model produces higher values and greater variability compared to the ODE.MLP and GR.MLP models. This likely results from the hybridization effect of the process-parameterization neural network ϕ_1, which incorporates neutralized rainfall to refine the physical equations in the ODE system. This creates a rainfall sensitivity in the transfer state, which is not accounted for in the classical GR and ODE structures.

Nevertheless, Figure 9 reveals that all three model structures exhibit relatively similar patterns in time-averaged transfer state maps, though with higher mean values (0.18 compared to 0.16 for ODE.MLP and 0.17 for GR.MLP) in the case of NODE.MLP. The maps of time-averaged production state for NODE.MLP and ODE.MLP display different patterns from those of GR.MLP. However, no evident differences emerge between ODE.MLP and NODE.MLP for both h_p and h_t. This is unsurprising as spatial hybridization effects are not expected since the process-parameterization neural network employs a simple MLP (MLP

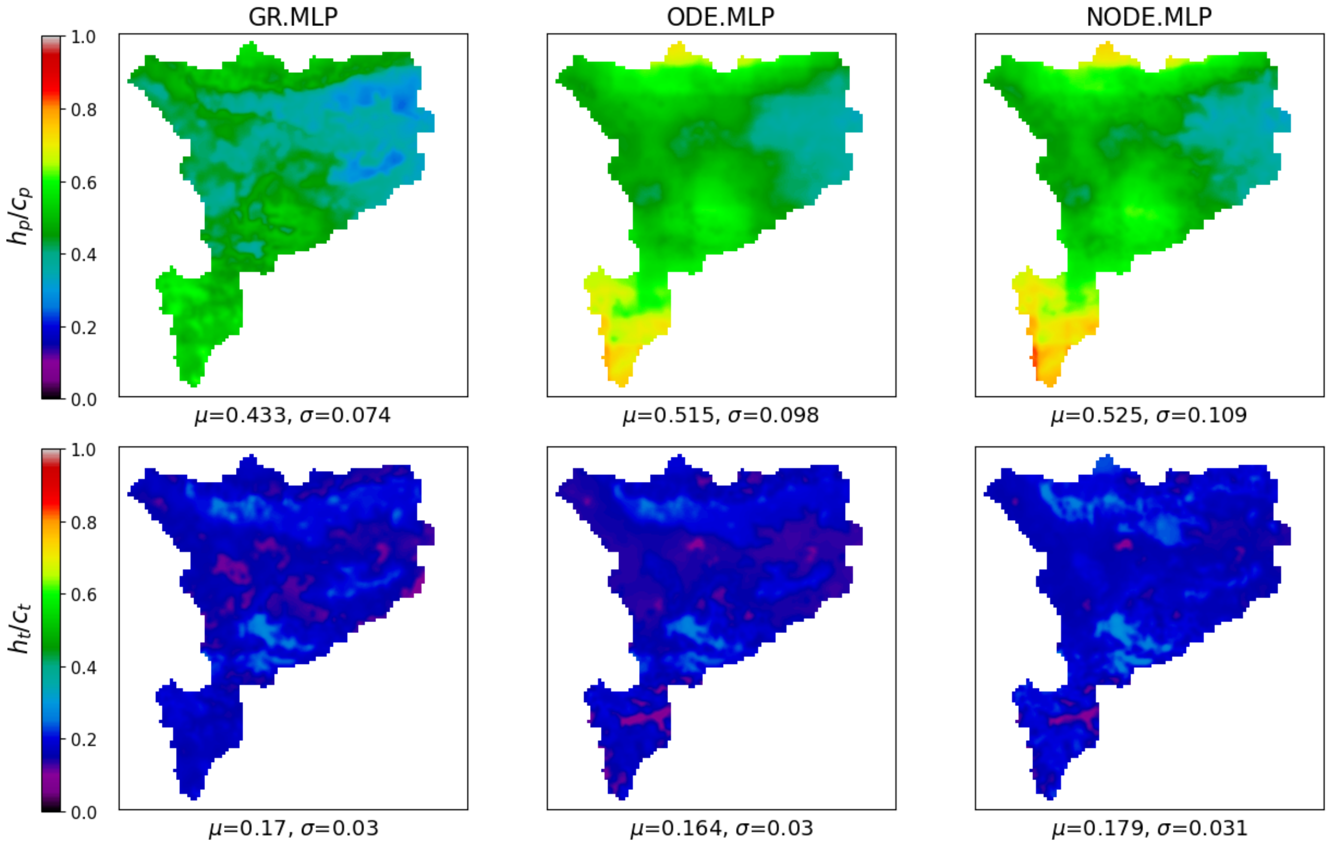

Figure 9. Maps of the time-averaged normalized production state $\overline{h_p(x)/c_p(x)}^t$ and transfer state $\overline{h_t(x)/c_t(x)}^t$ over the validation period P2, for the three model structures (GR, ODE, NODE), each employing an MLP-based regionalization mapping. For each map, μ and σ denote the spatial average and standard deviation.

of ϕ_1 and not MLP of the regionalization mapping ϕ_2) that does not account for or accounts less for spatial information. Future work could employ other types of ϕ_1 to explore this aspect.

Finally, Figure 10 illustrates the hybridization effect on runoff flow (lateral discharge for routing operator) by examining high-pass filtered lateral discharge from the most downstream gauge during the validation period. A 3-day cutoff frequency high-pass filter removes seasonal and long-term patterns to focus on flood event behavior. During major events, GR.MLP consistently produces lower lateral discharge compared to ODE.MLP. In certain situations, this results in an overestimation of flood magnitude with the ODE structure and an underestimation with the GR structure. The hybrid NODE structure effectively addresses this issue by refining internal water fluxes in the ODE system, resulting in moderate lateral discharge values that fall between those of ODE.MLP and GR.MLP. This effect is consistent with previously observed flood simulation results, demonstrating the improved hydrological response representation achieved through neural ODE integration.

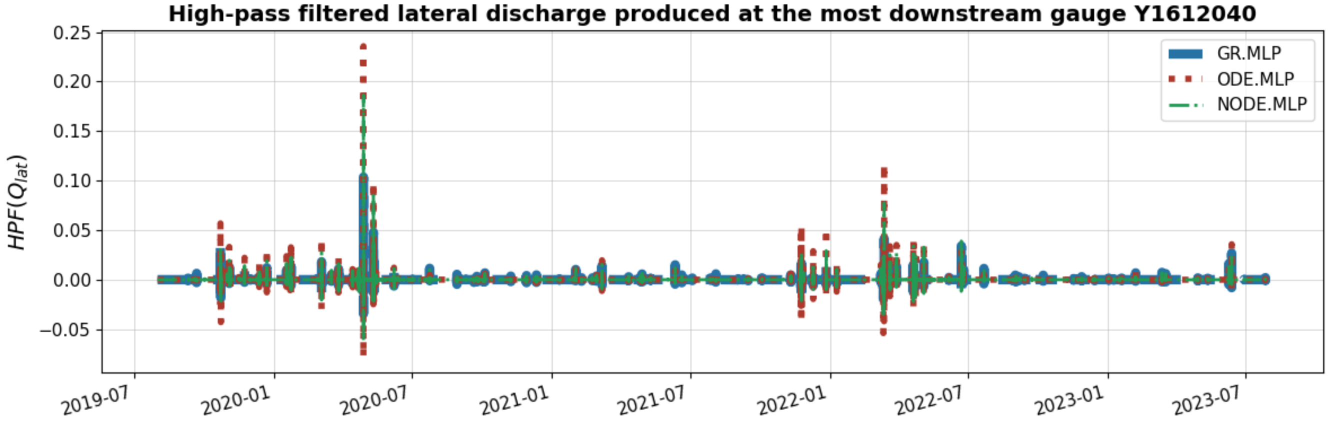

Figure 10. High-pass filtered lateral discharge from the most downstream gauge, using a 3-day cutoff frequency, to emphasize flood events during the validation period P2. The results are shown for three model structures (GR, ODE, NODE), each employing an MLP-based regionalization mapping.

4 General discussion on pure AI and hybrid modeling approaches

It has been nearly 80 years since Alan Turing first introduced the concept of a Turing machine, paving the way for the realization of thinking machines (Turing, 1950). Since then, scientists have made impressive efforts to simulate biological brain functions based on mathematical principles and the understanding of natural learning processes. AI models, with their generalization ability to learn multi-level abstractions from large datasets through backpropagation algorithms, have dramatically helped automate various tasks in scientific and engineering applications (LeCun et al., 2015).

By 2025, AI has become ubiquitous, often perceived as a “magical” tool capable of addressing numerous challenges. Although the results of its applications seem remarkable, the evolution of AI is deeply rooted in the advancement of computational power and the application of well-established mathematical principles. The foundational concepts of modern AI are grounded in linear algebra, probability (e.g., Bayesian inference), and control theory (Lions, 1971) (see Goodfellow et al. (2016) for fundamental concepts of modern AI). Many of the key algorithms and model architectures that form the core of modern AI were developed much earlier in the 20th century but did not achieve the widespread success we see today. The primary reason for this was the lack of high-quality, extensive data and sufficient computational power to train these models effectively at that time (LeCun et al., 2015).

The intelligence we commonly reference in modern AI systems is, indeed, primarily the ability to learn patterns provided by humans through data. In other words, current AI models do not possess the capability to explore patterns beyond the given data, or even when they do, such explorations are not considered significant discoveries if researchers cannot interpret them (LeCun, 2018). Physics, on the other hand, focuses on decoding phenomena using systems of mathematical equations, discovered through human intelligence and continually validated in pursuit of a unified theory of the universe.

https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

The question then arises: should we focus on employing complex AI models to partially or fully replace process-based models, or should we prioritize understanding the physical interpretability of hybrid approaches, starting with simpler AI architectures? There is no straightforward answer to this question. However, based on our knowledge of both disciplines, current AI models are not capable of fully replacing process-based models, given the inherent complexity of hydrological processes and uncertainties in modeling from limited observations. Full replacement might become feasible if at least the following two conditions are met:

- AI models become more intelligent, producing more interpretable results and demonstrating a comprehensive understanding of physics. However, this is not currently the case, as highlighted by LeCun (2022);
- We obtain nearly perfect and complete hydrological data, accounting for all uncertainties. This requires significant effort and remains a distant goal (Beven, 2019).

Given current advancements in both hydrological modeling and AI, we believe that a comprehensive understanding of complex hydrological processes through a stand-alone AI model is not yet feasible. Even as datasets become more extensive, they remain insufficient to fully capture all relevant physical processes due to observational limitations, biases, and the inherent complexity of hydrological systems (Beven, 2019). Researchers can easily fall into the trap of optimizing for impressive performance metrics on local datasets while failing to develop models that accurately represent underlying physical mechanisms (Reichstein et al., 2019). Instead, we should focus on hybrid approaches that seamlessly integrate AI models into hydrological processes to leverage AI capabilities while maintaining physical consistency and process understanding.

5 Conclusions

This study proposed a framework for integrating neural networks into ODEs of a process-based hydrological model. The hybrid ODE system, referred to as neural ODEs, incorporates a neural network that refines internal water fluxes and is governed by physical equations to describe the model's state dynamics. Moreover, the hydrological model includes a kinematic wave routing and a regionalization mapping. This mapping enables learning the transfer function between physical descriptors and spatially distributed conceptual parameters, as introduced in Huynh et al. (2024), which is now further enhanced by employing a CNN in addition to an MLP.

The proposed methods were tested on multiple catchments in the Aude River basin, demonstrating that increased complexity in model structures (from classical GR to ODE and neural ODE structures) and regionalization mapping (from uniform to MLP and CNN) leads to improved model calibration scores. Furthermore, models using CNN-based regionalization mapping generally exhibited smoother estimated parameter maps. This smoothness is a result of the convolution operation, which applies a relatively fine filter to the input descriptors. Evaluation scores from various validation scenarios suggest that the classical GR model using CNN (GR.CNN) and the neural ODE structure using MLP (NODE.MLP) demonstrated consistently strong and stable performance compared to other models. Analysis of state dynamics and runoff flow revealed the hybridization effect of the neural ODE structure, which modifies internal water fluxes to achieve more reliable streamflow simulations during flood events. This preliminary analysis examines recently proposed hybrid solvers for spatially distributed modeling, which could be further explored under a broader range of hydrological conditions, experimental hypotheses, and state-parameter analyses.

By leveraging AI capabilities within physically-based frameworks, we aim to develop more robust, interpretable, and generalizable models for hydrological processes. This approach not only enhances the accuracy of streamflow predictions but also provides deeper insights into the underlying physical mechanisms. Additionally, it employs AI capabilities to handle massive data within a fully distributed approach for flood modeling, thereby representing a significant advancement toward a robust hybrid AI approach that emphasizes physical understanding and interpretability.

Code and data availability. The source code of `smash`, Version 1.1, is available and preserved on multiple platforms: GitHub at https://github.com/DassHydro/smash/tree/v1.1.0, PyPI at https://pypi.org/project/hydro-smash/1.1.0, and Zenodo at https://doi.org/10.5281/zenodo.15498851 (Huynh et al., 2025a). The dataset supporting this study comprises preprocessed data sourced from SCHAPI-DGPR and Météo-France, and are available at https://doi.org/10.5281/zenodo.15315600 (Huynh, 2025). `smash` is released under the GPL-3 license and developed openly at https://github.com/DassHydro/smash. The documentation is accessible at https://smash.recover.inrae.fr.

Author contributions. NNTH: methodology and conceptualization, main developer of `smash` v1.1, main writing, numerical experiments, results analysis. PAG: methodology and conceptualization, co-developer of `smash` v1.1, manuscript review, results analysis, supervision and funding. FC: co-developer of `smash` v1.1. JM: conceptual discussions.

Competing interests. The authors declare that no competing interests are present.

Acknowledgements. The authors would like to acknowledge the support of funding from SCHAPI-DGPR, ANR grant ANR-21-CE04-0021-01 (MUFFINS project, “MUltiscale Flood Forecasting with INnovating Solutions”, https://muffins-anr-project.hub.inrae.fr), and NEPTUNE European project DG-ECO.

References

Beven, K.: How to make advances in hydrological modelling, Hydrology Research, 50, 1481–1494, https://doi.org/10.2166/nh.2019.134, 2019.

Beven, K.: Deep learning, hydrological processes and the uniqueness of place, Hydrological Processes, 34, 3608–3613, https://doi.org/10.1002/hyp.13805, 2020.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D.: Neural Ordinary Differential Equations, https://arxiv.org/abs/1806.07366, 2019.

Cho, K. and Kim, Y.: Improving streamflow prediction in the WRF-Hydro model with LSTM networks, Journal of Hydrology, 605, 127 297, https://doi.org/10.1016/j.jhydrol.2021.127297, 2022.

Colleoni, F., Garambois, P.-A., Javelle, P., Jay-Allemand, M., and Arnaud, P.: Adjoint-based spatially distributed calibration of a grid GR-based parsimonious hydrological model over 312 French catchments with SMASH platform, EGUsphere, 2022, 1–37, https://doi.org/10.5194/egusphere-2022-506, 2022.

Colleoni, F., Huynh, N. N. T., Garambois, P.-A., Jay-Allemand, M., Organde, D., Renard, B., De Fournas, T., El Baz, A., Demargne, J., and Javelle, P.: SMASH v1.0: A Differentiable and Regionalizable High-Resolution Hydrological Modeling and Data Assimilation Framework, EGUsphere, 2025, 1–36, https://doi.org/10.5194/egusphere-2025-690, 2025.

Feng, D., Fang, K., and Shen, C.: Enhancing Streamflow Forecast and Extracting Insights Using Long-Short Term Memory Networks With Data Integration at Continental Scales, Water Resources Research, 56, e2019WR026 793, https://doi.org/doi.org/10.1029/2019WR026793, 2020.

Feng, D., Liu, J., Lawson, K., and Shen, C.: Differentiable, Learnable, Regionalized Process-Based Models With Multiphysical Outputs can Approach State-Of-The-Art Hydrologic Prediction Accuracy, Water Resources Research, 58, e2022WR032 404, https://doi.org/10.1029/2022WR032404, 2022.

Ficchì, A., Perrin, C., and Andréassian, V.: Hydrological modelling at multiple sub-daily time steps: Model improvement via flux-matching, Journal of Hydrology, 575, 1308–1327, https://doi.org/10.1016/j.jhydrol.2019.05.084, 2019.

Frame, J. M., Kratzert, F., Raney II, A., Rahman, M., Salas, F. R., and Nearing, G. S.: Post-Processing the National Water Model with Long Short-Term Memory Networks for Streamflow Predictions and Model Diagnostics, JAWRA Journal of the American Water Resources Association, 57, 885–905, https://doi.org/10.1111/1752-1688.12964, 2021.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep Learning, MIT Press, http://www.deeplearningbook.org, 2016.

He, Q., Barajas-Solano, D., Tartakovsky, G., and Tartakovsky, A. M.: Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport, Advances in Water Resources, 141, 103 610, https://doi.org/10.1016/j.advwatres.2020.103610, 2020.

Höge, M., Scheidegger, A., Baity-Jesi, M., Albert, C., and Fenicia, F.: Improving hydrologic models for predictions and process understanding using neural ODEs, Hydrology and Earth System Sciences, 26, 5085–5102, https://doi.org/10.5194/hess-26-5085-2022, 2022.

Huynh, N. N. T.: Aude River basin (1km, 1h), https://doi.org/10.5281/zenodo.15315600, dataset, 2025.

Huynh, N. N. T., Garambois, P.-A., Colleoni, F., and Javelle, P.: Signatures-and-sensitivity-based multi-criteria variational calibration for distributed hydrological modeling applied to Mediterranean floods, Journal of Hydrology, 625, 129 992, https://doi.org/10.1016/j.jhydrol.2023.129992, 2023.

https://doi.org/10.5194/egusphere-2025-2797
Preprint. Discussion started: 7 July 2025

Huynh, N. N. T., Garambois, P.-A., Colleoni, F., Renard, B., Roux, H., Demargne, J., Jay-Allemand, M., and Javelle, P.: Learning Regionalization Using Accurate Spatial Cost Gradients Within a Differentiable High-Resolution Hydrological Model: Application to the French Mediterranean Region, Water Resources Research, 60, e2024WR037 544, https://doi.org/10.1029/2024WR037544, 2024.

Huynh, N. N. T., Colleoni, F., El Baz, A., Garambois, P.-A., Jay-Allemand, M., Renard, B., Akhtari, A., and Nguyen, N. B.: SMASH v1.1.0, https://doi.org/10.5281/zenodo.15498851, software, 2025a.

Huynh, N. N. T., Garambois, P.-A., Renard, B., Colleoni, F., Monnier, J., and Roux, H.: A Distributed Hybrid Physics-AI Framework for Learning Corrections of Internal Hydrological Fluxes and Enhancing High-Resolution Regionalized Flood Modeling, EGUsphere, 2025, 1–34, https://doi.org/10.5194/egusphere-2024-3665, 2025b.

Jay-Allemand, M., Javelle, P., Gejadze, I., Arnaud, P., Malaterre, P.-O., Fine, J.-A., and Organde, D.: On the potential of variational calibration for a fully distributed hydrological model: application on a Mediterranean catchment, Hydrology and Earth System Sciences, 24, 5519–5538, https://doi.org/10.5194/hess-24-5519-2020, 2020.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrology and Earth System Sciences, 22, 6005–6022, https://doi.org/10.5194/hess-22-6005-2018, 2018.

LeCun, Y.: The Power and Limits of Deep Learning, Research-Technology Management, 61, 22–27, https://doi.org/10.1080/08956308.2018.1516928, 2018.

LeCun, Y.: A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27, Open Review, 62, 1–62, 2022.

LeCun, Y., Bengio, Y., and Hinton, G.: Deep learning, Nature, 521, 436–444, https://doi.org/10.1038/nature14539, 2015.

Lions, J. L.: Optimal Control of Systems Governed by Partial Differential Equations, vol. 170 of *Grundlehren der mathematischen Wissenschaften*, Springer Berlin, Heidelberg, 1 edn., ISBN 978-3-642-65026-0, 1971.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S.: PyTorch: an imperative style, high-performance deep learning library, Curran Associates Inc., Red Hook, NY, USA, 2019.

Perrin, C., Michel, C., and Andrèassian, V.: Improvement of a parsimonious model for streamflow simulation, Journal of hydrology, 279, 275–289, https://doi.org/10.1016/S0022-1694(03)00225-7, 2003.

Piotte, O., Montmerle, T., Fouchier, C., Belleudy, A., Garandeau, L., Janet, B., Jauffret, C., Demargne, J., and Organde, D.: Les évolutions du service d'avertissement sur les pluies intenses et les crues soudaines en France, La Houille Blanche, 106, 75–84, https://doi.org/10.1051/lhb/2020055, 2020.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A.: Universal Differential Equations for Scientific Machine Learning, 2021.

Raissi, M., Perdikaris, P., and Karniadakis, G.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, 378, 686–707, https://doi.org/10.1016/j.jcp.2018.10.045, 2019.

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

Santos, L., Thirel, G., and Perrin, C.: Continuous state-space representation of a bucket-type rainfall-runoff model: a case study with the GR4 model using state-space GR4 (version 1.0), Geoscientific Model Development, 11, 1591–1605, https://doi.org/10.5194/gmd-11-1591-2018, 2018.

Shen, C.: A Transdisciplinary Review of Deep Learning Research and Its Relevance for Water Resources Scientists, Water Resources Research, 54, 8558–8593, https://doi.org/10.1029/2018WR022643, 2018.

Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., Baity-Jesi, M., Fenicia, F., Kifer, D., Li, L., Liu, X., Ren, W., Zheng, Y., Harman, C. J., Clark, M., Farthing, M., Feng, D., Kumar, P., Aboelyazeed, D., Rahmani, F., Song, Y., Beck, H. E., Bindas, T., Dwivedi, D., Fang, K., Höge, M., Rackauckas, C., Mohanty, B., Roy, T., Xu, C., and Lawson, K.: Differentiable modelling to unify machine learning and physical models for geosciences, Nature Reviews Earth & Environment, 4, 552–567, https://doi.org/10.1038/s43017-023-00450-9, 2023.

Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y., and Demir, I.: A comprehensive review of deep learning applications in hydrology and water resources, Water Science and Technology, 82, 2635–2670, https://doi.org/10.2166/wst.2020.369, 2020.

Song, Y., Knoben, W. J. M., Clark, M. P., Feng, D., Lawson, K., Sawadekar, K., and Shen, C.: When ancient numerical demons meet physics-informed machine learning: adjoint-based gradients for implicit differentiable modeling, Hydrology and Earth System Sciences, 28, 3051–3077, https://doi.org/10.5194/hess-28-3051-2024, 2024.

Te Chow, V., Maidment, D. R., and Mays, L. W.: Applied Hydrology, McGraw-Hill Series in Water Resources and Environmental Engineering, 1988.

Turing, A. M.: I.—COMPUTING MACHINERY AND INTELLIGENCE, Mind, LIX, 433–460, https://doi.org/10.1093/mind/LIX.236.433, 1950.

Wang, C., Jiang, S., Zheng, Y., Han, F., Kumar, R., Rakovec, O., and Li, S.: Distributed Hydrological Modeling With Physics-Encoded Deep Learning: A General Framework and Its Application in the Amazon, Water Resources Research, 60, e2023WR036 170, https://doi.org/10.1029/2023WR036170, 2024.

Xiang, Z., Yan, J., and Demir, I.: A Rainfall-Runoff Model With LSTM-Based Sequence-to-Sequence Learning, Water Resources Research, 56, e2019WR025 326, https://doi.org/10.1029/2019WR025326, 2020.

Yin, Y., Guen, V. L., Dona, J., de Bézenac, E., Ayed, I., Thome, N., and Gallinari, P.: Augmenting physical models with deep networks for complex dynamics forecasting, Journal of Statistical Mechanics: Theory and Experiment, 2021, 124 012, https://doi.org/10.1088/1742-5468/ac3ae5, 2021.