

Response to comments on Manuscript egusphere-2026-4655 (RC2)

Title: An Open-Source Python Framework for the Flexible Deployment and Advanced Applications of the Variable Infiltration Capacity (VIC) Model

Authors: Xudong Zheng, Dengfeng Liu*, Qiuhong Tang, Qiang Li, Haoyu Liao, Hao Wang

Manuscript ID: egusphere-2026-4655

Citation: <https://doi.org/10.5194/egusphere-2026-4655-RC2>

Reply on RC2:

Many thanks for taking the time and effort to review our paper. All comments from Referee #2 are addressed below with point-by-point responses.

For better readability, our replies are prefixed with “**R/**”. The original reviewer comments are prefixed with “**RC-n/**” and are presented in **bold**. The revisions to be added into the revised manuscript is highlighted in **red**, while unchanged text from the original manuscript is shown in **grey**. The important parts are highlighted in **blue**. The quoted content is displayed in *italics*. All references cited in this response are listed at the end of the reply.

Point-to-point response:

RC2-1/ The EVB framework addresses a real and important bottleneck in VIC-5 deployment. Its Python-based, object-oriented design with integrated MPR is well conceived, and the two case studies cover both large-sample and site-specific application scenarios. The main weaknesses lie in lack of comparison with existing tools and insufficient methodological transparency.

R/ We sincerely thank you for your careful and constructive review, and for recognizing the importance and relevance of our EVB framework for VIC-5 deployment. We also **acknowledge the limitations** you pointed out and greatly appreciate these valuable suggestions for improving our manuscript. We provide point-by-point responses to your comments below and have made every effort to address them through corresponding revisions. The corresponding revisions will be incorporated into the subsequent revised manuscript.

RC2-2/ Detailed Comments 1. There is no quantitative comparison with existing tools. The introduction criticizes VIC-ASSIST at some length, yet no functional or performance comparison is provided. A tool paper should address: How much faster is EVB than VIC-ASSIST? How much more flexible? What can EVB do that VIC-ASSIST cannot? A feature comparison matrix (model versions supported, data formats, calibration algorithms, parameter regionalization, parallel computing, etc.) is recommended. The positioning of EVB relative to alternatives such as the VIC Python driver ecosystem and SUMMA+mizuRoute also deserves brief discussion.

R/ Thank you for your valuable suggestion. We agree that a systematic comparison with existing VIC deployment tools, such as VIC-ASSIST, is helpful for clarifying the scope and contributions of the EVB framework within the current VIC modeling ecosystem.

However, we would like to clarify that **a direct quantitative comparison** of computational efficiency (i.e., “how much faster”) between EVB and VIC-ASSIST **may not provide a meaningful assessment**. This is because the two tools were developed for **different VIC versions, data structures, and computational workflows**, which can substantially influence the overall deployment efficiency. Specifically, (1) VIC-ASSIST was primarily designed for VIC-4 deployment, whereas EVB targets the latest VIC-5 framework; (2) VIC-ASSIST mainly relies on text-based input/output interfaces, while EVB adopts a NetCDF-based data interfaces; (3) the two tools employ different spatial data preprocessing and parameter estimation procedures (i.e., spatially uniform parameters in VIC-ASSIST versus MPR-based spatial parameters in EVB); (4) VIC-ASSIST relies on MATLAB and ArcGIS for Desktop, whereas EVB is developed as a Python-based open-source framework built upon publicly available Python packages without requiring third-party software dependencies. Therefore, a runtime comparison between them would inevitably be influenced by multiple aspects of the modeling workflow, making it difficult to isolate the efficiency of the deployment framework itself.

On the other hand, we would like to emphasize that **our intention is not to undermine VIC-ASSIST** or highlight its limitations. We highly appreciate its contribution to facilitating VIC deployment through a user-friendly GUI environment. The two tools address different needs within the VIC modeling ecosystem: VIC-ASSIST primarily emphasizes user-friendly GUI-assisted deployment, whereas EVB focuses on reproducibility, flexibility, and extensibility for advanced modeling workflows.

For these reasons, we do not perform a direct efficiency comparison between the two tools. Instead, following your suggestion, we **have added a feature comparison matrix** to the new Discussion section (Sect. 6.2) to clarify the differences between EVB and existing VIC deployment tools. To provide a more comprehensive overview of the current VIC deployment ecosystem, we also include **TONIC** and **CyberWater** in the comparison. Meanwhile, to maintain consistency and better clarify our intended message while avoiding potential misunderstandings, we have further revised the Introduction by **refining the description of VIC-ASSIST**, reducing the use of potentially critical wording.

Following your suggestion, we have also acknowledged the **workflow blueprint for coupled SUMMA and mizuRoute modelling** proposed by **Knoben et al. (2022)** and the **CWARHM initiative**. Their emphasis on reproducibility, separation of concerns, modular design, layered organization, and deployment traceability **provides useful context for the design of EVB**. We have therefore expanded the discussion of these related efforts in the Introduction and Discussion (Section 6.2) sections, emphasizing their relevance to the design philosophy of the EVB framework. We also briefly discuss the VIC Python driver ecosystem as a related development with implications for the future compatibility and development of EVB.

We believe these revisions help address your concerns regarding the positioning of EVB and provide a more balanced description of related VIC deployment efforts. For your convenience in reviewing these changes, we have provided the revised Introduction section and the newly added discussion below.

1 Introduction

... ..

However, deploying VIC entails a series of laborious and technically complex tasks, including model configuration, input data preparation, and parameter calibration. These requirements can limit its broader use and the development of more advanced applications. The widespread adoption of the Soil and Water Assessment Tool (SWAT) illustrates the value of accessible deployment tools: its intuitive interface and integration with geographic information systems (GIS)

have helped users overcome practical implementation challenges (Qiu et al., 2017). VIC presents a different situation. It is distributed primarily as source code, leaving users to preprocess data themselves and rely on external tools to extract spatial information. As a result, configuring VIC across computing environments and spatial scales remains a largely manual process.

Workflow-based deployment provides a structured and transparent means of enhancing the reproducibility of hydrological modelling studies, while also improving modelling efficiency by reducing the effort required for otherwise cumbersome deployment procedures (Knoben et al., 2022). These benefits, which are particularly valuable for experiments spanning multiple research domains or spatiotemporal scales, allow researchers to devote more attention to process understanding and scientific questions rather than implementation details (Tang et al., 2025). However, workflow-based deployment frameworks specifically designed for VIC remain scarce. One notable exception is the VIC-Automated Setup Toolkit (VIC-ASSIST), developed by Wi et al. (2017) in the MATLAB environment. Through a carefully designed graphical user interface, VIC-ASSIST enables users with limited programming and modelling experience to deploy the VIC model rapidly for a specified study region. Despite its ease of use, the tool relies on proprietary third-party software, such as MATLAB and ArcGIS, uses several fixed configuration settings, including a restriction to three soil layers, and supports only VIC-4. These limitations reduce its portability, flexibility, and applicability for advanced modelling applications based on the more recent VIC-5 version. Another example is TONIC, a Python-based processing toolbox developed by the VIC development group (<https://github.com/UW-Hydro/tonic>). TONIC was intended to provide a broader set of tools for VIC, but its available functionality remains limited to a small number of tasks, such as plotting and data format conversion, and the toolbox has not been updated for several years. It therefore does not currently provide a useful workflow for reproducible VIC deployment.

More recently, CyberWater has attracted attention as a general graphical workflow environment for VIC applications (Salas et al., 2020; Bieger et al., 2025). Developed as an open data and open model cyberinfrastructure, CyberWater uses the visual workflow system VisTrails to integrate heterogeneous data sources and models through configurable graphical modules. Users can construct VIC workflows by connecting data and model agents, with minimal programming required. It is among the most fully developed VIC deployment frameworks currently available and has been introduced through several training workshops, which has helped establish a user base within the VIC community. However, the high level of encapsulation inherent in this software-based design can constrain extensions for more advanced modelling applications, such as spatial parameter estimation through regionalization. Under its current design, users can modify or extend CyberWater only by inserting individual pieces of Python code through the PythonSource module, with limited access to the underlying deployment components and workflow structure. The need to install and maintain the CyberWater environment may also impose additional learning and storage requirements. A complementary code based framework would therefore remain useful, particularly for advanced users who require direct control over deployment components or prefer to work without installing an additional software platform.

In the absence of a shared workflow for flexible and reproducible VIC deployment, differences in data processing and file organization can hinder the effective application of the model and make it difficult for users to exchange and build on one another's work. To address this need, we introduce Easy VIC Build (EVB), an open-source Python framework for deploying VIC-5 that integrates data preparation, model configuration, and file organization within a common workflow. It is implemented entirely in Python, requires no proprietary software, and can be extended through the broader Python ecosystem, for example, by incorporating packages that implement various optimization algorithms. EVB adopts a loosely coupled, object-oriented design based on the separation of concerns (SoC) principle, consistent with the workflow blueprint proposed by Knoben et al. (2022). By preserving the modularity of individual components, this design provides flexibility to extend the framework further by modifying or replacing any step in the deployment workflow without disrupting the overall architecture. It also enables EVB to accommodate data from diverse sources and in a wide range of formats. Furthermore, EVB incorporates two advanced model deployment methods, namely

Multiscale Parameter Regionalization (MPR; Samaniego et al., 2010; Mizukami et al., 2017) and the General Unit Hydrograph (GUH; Guo, 2022a). These capabilities broaden the applicability of EVB and provide a basis for more advanced VIC modelling applications. We evaluate EVB across modelling applications of varying complexity and basin configurations, focusing on its deployment efficiency and adaptability. The resulting reduction in model deployment effort may lower technical barriers to VIC use and promote greater consistency in modelling practices within the VIC community. In this respect, EVB responds to the CWARHM (Community Workflows to Advance Reproducibility in Hydrologic Modelling) initiative by translating its principles into a VIC specific framework for reproducible model deployment (Knoben et al., 2022).

The remainder of this paper is organized as follows. Section 2 describes the design and architecture of the EVB framework, together with its environmental requirements. Section 3 presents its main functionalities and the workflow of VIC model deployment. Section 4 provides a description of the case basins and data collection, while Sect. 5 demonstrates several modelling applications based on EVB. Section 6 discusses the opportunities and challenges of spatial parameter estimation in distributed hydrological models and examines the positioning and scope of EVB. Finally, Sect. 7 presents the conclusions.

6.2 The positioning and scope of EVB

Positioning EVB within the existing VIC deployment ecosystem helps clarify its role and scope. Given the limited number of deployment tools available for VIC, we selected three representative examples, VIC-ASSIST, TONIC, and CyberWater, each briefly introduced in the Introduction, for comparison with EVB. We compiled a feature matrix covering supported VIC versions, implementation languages and distribution forms, approaches to parameter specification and regionalization, workflow reproducibility, and extensibility (Table 5). All three tools in the comparison are freely available, and several are distributed under open-source licenses (e.g. MIT license). This accessibility is consistent with the recommendation of Knoben et al. (2022) to use permissively licensed workflow components that support reuse and extension. However, free access to a deployment tool does not eliminate software dependencies. As noted earlier, VIC-ASSIST depends on MATLAB and ArcGIS, whose commercial licensing requirements may limit its use where these software packages are unavailable.

Functionally, TONIC has the narrowest scope among the four tools. It provides a small set of utility scripts, mainly for NetCDF and ASCII conversion and data visualization. It therefore does not constitute an integrated deployment workflow. VIC-ASSIST focuses on the rapid deployment of VIC 4 and simplifies operation by fixing several model choices, such as data sources and parameter calibration settings. This design is more suitable for new users and educational applications (Wi et al., 2017). By comparison, CyberWater combines a graphical interface with data and model agents, offering a practical balance between ease of use and functional breadth. It is accessible to users with limited programming experience and provides more configuration options than VIC-ASSIST. However, its encapsulated software architecture makes the workflow less straightforward to extend. For example, incorporating a new optimization algorithm may leave users or developers to adapt or reimplement the algorithm in the format expected by the workflow, rather than directly reusing an existing implementation from the broader Python ecosystem. EVB places greater demands on users' Python programming experience. In return, it provides direct control over every stage of the deployment workflow, from data preparation and parameter specification to calibration and parallel execution, supporting a broader range of advanced modelling applications (Table 5). Given the rapid development of large language model based programming tools, such as Codex, we believe that this programming requirement is unlikely to constitute a major barrier to the use of EVB.

Knoben et al. (2022) developed a workflow blueprint for the coupled SUMMA and mizuRoute modelling system as part of the CWARHM initiative. The blueprint has provided a reference for workflow based modelling in the Earth system science community by setting out guidelines for improving reproducibility through four key properties: separability, clarity, modularity, and traceability. Within this broader context, several aspects of the EVB design are consistent with

the guidelines. First, EVB follows the separation of concerns (SoC) principle through a modular, loosely coupled, object-oriented design, allowing individual components to be modified or replaced independently. For example, the two basin setups examined here use CONUS SOIL and SoilGrids1km, respectively. The two datasets represent soil profiles using 11 layers spanning 0–250 cm and 6 layers spanning 0–200 cm. During parameter generation, EVB calls the same buildParameter function with two subclasses of SoilLayerResampler, one for each dataset. Each subclass requires only minimal overrides of selected methods, allowing the same function to accommodate both layer schemes without separate construction functions. This degree of component flexibility is not readily available in the other three tools. Consistent module interfaces also support workflow reproducibility. When Model A and Model B use the same data source, the dpc class can reuse the relevant modules within extractData_func without creating conflicts with other methods. Second, a structured directory design facilitates the exchange of workflows within the modelling community. Applications deployed with EVB follow a common directory structure managed by the Evb_dir class (Fig. 1c), allowing users to inspect and rapidly reuse existing studies. This design provides transparency while retaining flexibility for user defined extensions, a combination that is often difficult to achieve in graphical workflow systems. Third, EVB provides flexibility at two levels: runtime configuration of model options and extension beyond the original application scope. Its ability to combine alternative modelling decisions was demonstrated in our previous study (Zheng et al., 2026a), in which flexible deployment enabled eight cases with different combinations of spatial discretization and calibration configuration. By contrast, deployment frameworks that do not expose their underlying code may offer only limited scope for extension, with further development often depending primarily on the original developers. Taken together, these features position EVB to improve transparency within the VIC modelling community, facilitate the exchange of computational practices, and reduce duplicated effort across the broader scientific community. By making deployment procedures easier to inspect, reproduce, and extend, EVB may also accelerate the iterative development and refinement of VIC.

Last but not least, the current scope of EVB warrants further consideration. Its reliance on GeoDataFrame structures may become inefficient for very large domains at continental or global scales, where chunked and parallel processing with Dask (<https://docs.dask.org>) or comparable tools may be needed. In addition, the transferability of MPR derived parameter estimates has so far been evaluated over a limited range of spatial scales, and broader testing is therefore needed. Future development will focus on four directions: (1) introducing a standardized data layer between preprocessing stages, thereby allowing framework specific intermediate files to be replaced with interoperable formats, notably GeoTIFF and NetCDF; (2) developing interfaces for conversion between VIC versions to improve backward compatibility; (3) implementing vector-based modelling units to provide more flexible spatial representations; and (4) maintaining consistency with ongoing VIC development, particularly its Python driver, which is currently experimental and remains under development. Future integration with this driver could broaden the scope of EVB, improve compatibility with VIC, and contribute to the continued development of the VIC modelling ecosystem.

Table 5. Feature comparison of EVB with existing VIC deployment tools.

Feature	VIC-ASSIST	TONIC	CyberWater	EVB
Supported VIC version	VIC-4.2	VIC-4	VIC-4 and VIC-5	VIC-5
Implementation language	Matlab and Python	Python	Python (VisTrails); external model executables	Python
Distribution or access form	MATLAB GUI	Python package	Installable graphical workflow platform	Python package
Software required	MATLAB, ArcGIS, Cygwin	Python	Python, VisTrails, Java, GRASS GIS or Docker	Python
License	Not specified	MIT license	Installer license agreement required	MIT license
Supported data source	Limited predefined sources (bundled subfolders)	Not specified	Data agents for sources including NASA, ERA5, USGS, and HydroShare	Any source (extensible through overridden class methods)

Supported data formats	ASCII and NetCDF	ASCII and NetCDF	NetCDF, text-based tables, and raster data; format support varies by module	Any format (extensible through overridden class methods)
Spatially explicit parameter fields/regionalization	Optional semi-distributed (uniform within each subbasin)	Not supported	Not documented	Multiscale parameter regionalization
Free parameter	Fixed 15 runoff generation, snow, and routing parameters	Not specified	Under development (user defined)	Any VIC parameter (extensible through dedicated modules)
Parameter calibration	Single-objective GA (outlet streamflow NSE/KGE; optional multi-site pooling)	Not supported	Under development	Single- and multi-objective calibration; user defined objectives
Workflow reproducibility	Limited (GUI-driven execution; few shareable intermediate files)	Standalone scripts (no integrated deployment workflow)	Tracked and shareable (requires local reconfiguration)	Tracked and shareable (modular Python workflow with explicit intermediate files and logging)
Flexibility/Extensibility	Not documented; likely requires manual MATLAB source edits	Individual scripts modifiable	Limited (small Python snippets via PythonSource module)	High (modular classes and methods can be modified or replaced)
Parallel computing	VIC and RVIC parallelization	Not supported	Local or remote HPC execution	Any deployment component can be parallelized

*Note: GUI, graphical user interface; GA, genetic algorithm; MIT, Massachusetts Institute of Technology License; HPC, high performance computing; GRASS GIS, Geographic Resources Analysis Support System Geographic Information System; Docker, a containerization platform.

RC2-3/ 2. MPR transfer functions are not specified; methodological transparency is insufficient. MPR is presented as a core EVB feature, yet the mathematical form of the transfer functions is entirely absent — how soil properties map to Ks via g-parameters, how topography influences b_{infiltr}, whether vegetation parameters go through MPR or direct resampling, etc. Open-source code does not substitute for methodological transparency in the paper. Additionally, the selection of the 15-parameter reduced calibration set rests on two literature citations without an independent sensitivity analysis in this study.

R/ We thank you for identifying shortcomings in our methodological description, particularly the lack of explicit MPR transfer functions. This omission reduced the transparency of the MPR implementation and could potentially confuse readers. To address this issue, we have **added a summary table of the transfer functions in Appendix A**, explicitly specifying the predictor–parameter relationships for all model parameters.

In addition, we have **standardized the notation** of all model parameters and predictor variables and carefully revised the corresponding descriptions throughout the manuscript to ensure consistency and clarity. For your convenience, the newly added table of transfer functions is also provided below.

Table A2. Summary of the transfer functions and their formulation sources for VIC-specific parameterization in EVB.

Parameter	Description	Unit	Predictors	Transfer function	Reference
$b_{infiltr}$	The exponent of the variable infiltration capacity curve	[1]	Standard deviation of elevation (σ_{ele}), g_4 , g_5	$b_{infiltr} = \frac{\log \sigma_{ele} - g_4}{\log \sigma_{ele} + 10g_5}$	Dumenil and Todini (1992); Hurk and Viterbo (2003); Balsamo et al. (2009)
K_s	Saturated hydrologic conductivity	[mm/day]	Sand content fraction (f_{sand}), clay content fraction (f_{clay}), g_6 , g_7 , g_8	$K_s = 10^{g_6 + g_7 f_{sand} + g_8 f_{clay}} \times 25.4 \times 24$	Cosby et al. (1984)
θ_s	Saturated water content	[m ³ /m ³]	f_{sand} , f_{clay} , g_9 , g_{10} , g_{11}	$\theta_s = (g_9 + g_{10} f_{sand} + g_{11} f_{clay}) / 100$	Cosby et al. (1984)

Parameter	Description	Unit	Predictors	Transfer function	Reference
ψ_s	Saturation matric potential	[kPa]	f_{sands} , clay content fraction (f_{silt}), g_{12} , g_{13} , g_{14}	$\psi_s = -10^{g_{12} + g_{13}f_{sands} + g_{14}f_{silt}} \times 0.0980665$	Cosby et al. (1984)
b_{ret}	Slope of Campbell's retention curve in log space	[1]	f_{sands} , f_{clay} , g_{15} , g_{16} , g_{17}	$b_{ret} = g_{15} + g_{16}f_{clay} + g_{17}f_{sands}$	Cosby et al. (1984)
b	Exponent in Campbell's equation for hydraulic conductivity	[1]	b_{ret} , g_{18} , g_{19}	$b = g_{18} + g_{19}b_{ret}$	Campbell (1974)
θ_{fc}	Field capacity	[m ³ /m ³]	θ_s , field-capacity matric potential (ψ_{fc}), ψ_s , b_{ret} , g_{20}	$\theta_{fc} = g_{20}\theta_s \left(\frac{\psi_{fc}}{\psi_s}\right)^{\frac{1}{b_{ret}}}$	Campbell (1974)
D_4	Exponent of the nonlinear part of the outflow curve	[1]	g_{21}	$D_4 = g_{21}$	Mizukami et al. (2017)
D_1	Linear reservoir coefficient	[day ⁻¹]	K_s , mean slope angle (S), normalization factor of soil moisture (S_f) set to one, g_{22}	$D_1 = \frac{K_s S}{10^{g_{22}} \times S_f}$	Mizukami et al. (2017)
D_2	Nonlinear reservoir coefficient	[day ^{-D₄}]	K_s , S , S_f , g_{23}	$D_2 = \frac{K_s S}{10^{g_{23}} \times S_f^2}$	Mizukami et al. (2017)
D_3	Soil moisture at which baseflow transition occurs from linear to nonlinear	[mm]	θ_{fc} , layer thickness of the third soil layer (h_3), g_{24}	$D_3 = g_{24}\theta_{fc}h_3$	Mizukami et al. (2017)
d_p	Soil thermal damping depth	[m]	g_{25}	$d_p = 4g_{25}$	VIC documentation
ψ_b	Bubbling pressure of soil	[cm]	b , g_{26} , g_{27}	$\psi_b = g_{26}b + g_{27}$	Schaperow et al. (2021)
f_{quartz}	Quartz content of soil	[1]	f_{sands} , g_{28}	$f_{quartz} = g_{28}f_{sands}/100$	Peters-Lidard et al. (1998)
ρ_b	Bulk density of soil layer	[kg/m ³]	Reference bulk density in soil dataset ($\rho_{b,ref}$), g_{29}	$\rho_b = g_{29}\rho_{b,ref}$	Empirical adjustment
ρ_s	Soil particle density	[kg/m ³]	g_{30} , g_{31} , g_{32}	$\rho_{s,i} = 2685g_{31}, i = 0,1,2$	VIC documentation
θ_c	Fractional soil moisture content at the critical point	[1]	θ_{fc} , θ_s , g_{33}	$\theta_c = \frac{g_{33}\theta_{fc}}{\theta_s}$	Empirical adjustment
θ_{wp}	Wilting point	[m ³ /m ³]	θ_s , b_{ret} , ψ_s , g_{34}	$\theta_{wp} = g_{34}\theta_s \left(-\frac{1500}{\psi_s}\right)^{\frac{1}{b_{ret}}}$	Campbell (1974); Mizukami et al. (2017)
f_{wp}	Fractional soil moisture content at the wilting point	[1]	θ_{wp} , θ_s , g_{35}	$f_{wp} = \frac{g_{35}\theta_{wp}}{\theta_s}$	Mizukami et al. (2017)
z_{0s}	Surface roughness of bare soil	[m]	g_{36}	$z_{0s} = 0.001g_{36}$	Schaperow et al. (2021)
$z_{0,snow}$	Surface roughness of snowpack	[m]	g_{37}	$z_{0,snow} = 0.0005g_{37}$	Schaperow et al. (2021)
v	Average flow velocity within each grid	[LT ⁻¹]	Accumulated upstream area (acc), S , g_{38} , g_{39} , g_{40}	$g_{38}acc^{g_{39}}S^{g_{40}}$	K et al. (2005); Duc et al. (2007); Chen et al. (2019)
D	Average diffusion coefficient within each grid	[L ² T ⁻¹]	Flow distance within each grid (d), v , g_{41}	$g_{41}dv$	Yang et al. (2016)

*Note: Units are S (%), h_3 (m), acc (km²), and d (m). ψ_{fc} is set to -33, -20, and -10 kPa when $f_{sand} \leq 50\%$, $50\% < f_{sand} \leq 70\%$, and $f_{sand} > 70\%$, respectively. The source of VIC documentation is: <https://vic.readthedocs.io/en/master/>.

Regarding the selection of the 15 free parameters, we clarify that this decision was not based solely on the two cited studies. Rather, **it was informed by a broader body of research and modelling practice**, including work by the VIC model developers and sensitivity analyses conducted for VIC applications in different regions worldwide. To make the basis for this selection transparent, we reproduce several relevant passages from the original studies below.

Liang et al. (1994; one of the foundational VIC papers, cited over 3,900 times):

“Among the hydrological parameters, only three (b_i , D_s , W_s) would best be estimated using streamflow data if they are available...The other hydrological parameters can be estimated using, for instance, soil characteristics.”

Gao et al. (2010; a representative book on the principles and implementation of the VIC model):

“6.5 Calibration ...The usual implementation approach (see e.g. Nijssen et al. 1997) involves calibration of six parameters: a) the infiltration parameter (b_i), which controls the partitioning of rainfall (or snowmelt) into infiltration and direct runoff (a higher value of b_i gives lower infiltration and yields higher surface runoff); b) D_2 and D_3 , which are the second and third soil layer thicknesses (D_1 , the top soil layer depth, is usually specified a priori) and affect the water available for transpiration and baseflow respectively (thicker soil depths have slower runoff response — baseflow dominated — with higher evapotranspiration, but result in longer retention of soil moisture and higher baseflow in wet seasons); c) D_{smax} , D_s , and W_s , which are baseflow parameters and also are estimated via calibration. D_{smax} is the maximum baseflow velocity, D_s is the fraction of maximum baseflow velocity, and W_s is the fraction of maximum soil moisture content of the third soil layer at which non-linear baseflow occurs. These three baseflow parameters determine how quickly the water stored in the third soil layer is evacuated as baseflow (Liang et al. 1994). The three baseflow parameters and the third soil layer depth (d_3) (Nijssen et al., 2001a, Su et al., 2005) are used with only minor adjustment during the calibration, while the infiltration parameter (b_i) and the second soil depth (d_2) are targeted for intensive calibration. Parameters b_i and d_2 are calibrated independently.”

Gou et al. (2020; sensitivity-analysis-based VIC calibration in China):

“Even though the VIC model has 46 or more tunable parameters (Bennett et al., 2018), we chose only 13 parameters for this study, because the values of these 13 parameters are typically subject to calibration rather than direct measurement.”

“We found that three parameters—the infiltration parameter (B) and two of the soil depth parameters (D_1 , D_2)—are highly sensitive in most basins, while other parameter sensitivities are strongly related to the dynamic environment of the basin.”

Gu et al. (2021; multi-objective calibration of the VIC model using a stepwise surrogate model):

“Generally, the parameters selected for calibration are the thickness of the second and third soil moisture layers (d_2 , d_3), the fraction of maximum soil moisture where nonlinear baseflow occurs (W_s), the maximum velocity of baseflow (D_{smax}), the fraction of D_{smax} where nonlinear baseflow begins (D_s), and the variable infiltration curve parameter (b_{inf}) (Lohmann et al. 1996; Xie et al. 2007; Mishra et al. 2010; Voisin et al. 2011; Lu et al. 2013; Zhao et al. 2013; Kang et al. 2014; Hengade & Eldho 2016; Liu et al. 2017).”

The studies use different notation but identify **a broadly consistent set of sensitive VIC parameters**. This agreement reflects the model's underlying formulations, particularly the variable infiltration capacity curve, the Xinanjiang saturation excess runoff formulation, and the ARNO baseflow function. We selected 15 MPR g-parameters to define transfer functions for the runoff parameters listed in Table A2 (unrevised manuscript). Calibration adjusts these g-parameters, while the transfer functions use them to estimate the same sensitive VIC parameters identified in previous studies.

We agree that a sensitivity analysis of the g-parameters could provide additional insight. However, it would involve a higher-dimensional parameter space and could make parameter identifiability more difficult to assess. Because the sensitivity of the underlying VIC parameters has already been examined in the studies cited above, we consider a further sensitivity analysis unnecessary and beyond the scope of this study. In addition, **our earlier EVB-based modelling study** assessed the adequacy of the selected 15 parameters for VIC calibration and **achieved good simulation performance**, providing further support for their selection (Zheng et al., 2026). Given EVB's flexible and extensible design, any VIC parameters can be adjusted and freely combined through the parameter_set module, allowing this question to be examined in future studies.

To clarify the basis for our parameter selection, we have revised the explanation in the Methods section and added further references, as detailed below.

5.1 VIC Model deployment

...**A substantial body of research (Liang et al., 1994; Gou et al., 2020; Gu et al., 2021; Cortés-Salazar et al., 2023; Yeste et al., 2024), together with widespread VIC modelling practice, supports the selection of this reduced parameter set as comprising parameters commonly identified as sensitive.**

RC2-4/ 3. Two-year warm-up periods may be short for a distributed model with deep soil storage and baseflow reservoirs. More discussion of this choice would be helpful.

R/ Thank you for your suggestion. Accordingly, we have **added a discussion in Sect. 5.1** to explain the choice of the two-year warm-up period. The discussion considers evidence from previous studies, established modelling practice, practical constraints related to data availability, and the the observed stability of simulated runoff during the early simulation period. For your convenience, the added text is provided below.

5.1 VIC Model deployment

...

Both basin setups used a two-year warm-up period to allow the model states, particularly soil moisture, to reach stable initial conditions. This period length was guided by previous studies and established modelling practice. For example, Miao et al. (2020) found that soil moisture at nine representative grid cells had largely reached equilibrium after two years when developing a VIC based land surface hydrological dataset for China. Several other studies have likewise used warm-up periods of one to three years (Nicolle et al., 2014; Qiu et al., 2017; Mei et al., 2023). Drawing on this evidence and considering data availability, we selected a two-year warm-up period to maintain a practical balance among the warm-up, calibration, and validation periods. Simulated runoff remained relatively stable during the early part of the subsequent simulation period, with no pronounced extreme values, further supporting this choice.

RC2-5/ 4. The synthetic-rainfall parameter validation is somewhat circular: under homogeneous forcing and initial conditions, spatial variability is purely parameter-driven.

R/ We understand the concern raised by this point and agree that, because the synthetic rainfall and initial conditions are spatially uniform, the spatial variation in simulation is generated entirely by the parameter fields. Thus, this experiment cannot independently validate the parameter estimates themselves.

Our purpose, however, **was not** to demonstrate a **direct runoff–parameter correspondence**, which is inherent in the experimental design, but to examine **whether the parameter driven runoff patterns are consistent with general hydrological process understanding (expectations)** and therefore provide an indirect check of their plausibility.

For example, general understanding of hillslope runoff generation suggests that relatively flat areas may produce less surface runoff than steeper areas. We therefore examined whether this contrast could emerge in simulations driven entirely by the parameter fields (Fig. 8a).

Viewed in this way, the experiment highlights **the advantage of MPR in integrating land surface characteristics to spatially varying model parameters**. This information would be lost under a spatially uniform parameterization, a common strategy in previous VIC modelling studies, because the same parameter values are applied across all terrain conditions and therefore cannot represent terrain related differences independently of spatial variations in the forcing data.

Thus, we consider this assessment meaningful within its intended scope. To clarify this point, we revised the relevant explanation in Sect. 5.3.2 and added a dedicated discussion of spatial parameter estimation in Sect. 6.1. For ease of review, the revised text is provided below.

5.3.2 Physical plausibility of spatial parameter patterns

...

In this experiment, spatial variability arises solely from the model parameters fields., and thus their physical plausibility can be verified through the relationship between simulation results and spatial parameter patterns. We therefore assess their physical plausibility by examining whether the resulting parameter-driven runoff patterns are consistent with general hydrological expectations.

...

This pattern is physically plausible, as the general understanding of hillslope hydrological processes suggests that relatively flat areas tend to produce less runoff than steeper mountainous regions.

6.1 Opportunities and challenges of spatial parameter estimation in distributed hydrological models

The rapid growth of hydrological big data has enabled substantial advances in hydrological modelling, particularly in the development of increasingly sophisticated distributed models (Argentin et al., 2025). By explicitly representing the spatial variability of hydrological processes, distributed models can resolve the spatial patterns of hydrological states and fluxes that are unavailable from their lumped counterparts, offering valuable insights for water resources management and flood and drought risk assessment. For example, in this study we found that baseflow was generally more abundant in the southern part of the JRB, indicating potentially greater hydrological capacity to support afforestation in these areas (Fig. 9c). In contrast, the relatively lower baseflow in the northern part suggests that afforestation need to be implemented more cautiously to avoid increasing pressure on local water availability. In this context, integrating the increasingly available data sources for spatial parameter estimation presents new opportunities

for fully realizing the potential of distributed hydrological modelling.

Several fundamental challenges hinder the effective estimation of spatially distributed parameters. First, the historical reliance on the concept of “effective parameters” in hydrological modelling has resulted in many models retaining a large number of free parameters that require calibration and are implicitly assumed to be spatially homogeneous, limiting their ability to explicitly represent spatial heterogeneity. Second, introducing spatially varying parameters into traditional calibration frameworks substantially increases the dimensionality of the optimization problem, while the available observations are often insufficient to constrain such high-dimensional parameter spaces, leading to equifinality and ill-posedness. Finally, direct parameter estimation approaches generally require extensive prior information describing the relationships between parameters and environmental characteristics, while establishing such relationships requires extensive experimental investigations and coordinated efforts across multiple disciplines. Against this background, many distributed hydrological models still use spatially uniform parameters to maintain the number of free parameters at a level that can be reliably constrained through calibration, imposing substantial limitations on their capability to represent spatial heterogeneity. Our previous studies have shown that such simplifications create spatial trade-offs that undermine the physical realism of simulations and limit predictive performance (Zheng et al., 2026a).

To address these challenges, several strategies have been proposed for estimating spatially distributed parameters. A straightforward approach is to aggregate the modelling domain into spatial units, such as sub-basins, and independently calibrate parameters for each unit while assuming spatial homogeneity within it. This provides an intermediate representation between fully lumped and grid-based parameterizations, with implementations including spatial clustering (Huynh et al., 2023), representative grid sampling (Sun et al., 2021), and hydrological response units (HRUs). Nevertheless, the number of calibration parameters still increases with the number of spatial units, leaving the parameter-dimensionality problem largely unresolved. Another class of approaches can be broadly termed scaling methods, which infer parameter values at one spatial scale and relate them to hydrological processes or parameters at another through scale-invariant or explicitly scale-dependent relationships (Viney et al., 2004). For example, Barrios and Frances (2012) demonstrated the feasibility of estimating mesoscale effective parameters by aggregating microscale simulation outputs and solving the inverse problem. Regionalization approaches further extend this concept and exploit similarity principles (Bao et al., 2012), establishing relationships between catchment characteristics and parameters in data-rich regions (donors) and transferring them to data-scarce or ungauged regions (receptors). More importantly, if such relationships are applied consistently across all modelling grids, they can provide an effective pathway for spatial parameter estimation.

Within the regionalization framework, MPR offers several advantages over conventional approaches. As mentioned in Sect. 3.1, MPR establishes environment–parameter relationships using transfer functions informed by existing literature. Compared with standard regionalization (SR) methods based on empirical regression or similarity measures derived from spatial distance or physical attributes, this approach allows prior knowledge from multiple disciplines to be explicitly incorporated into parameter estimation, thereby improving the plausibility and reliability of spatial parameter estimates (Mizukami et al., 2017). Meanwhile, the limited number of adjustable parameters in transfer functions maintains the parameter dimensionality within a practically calibratable range. In terms of cross-scale issues, MPR first derives high-resolution parameter fields and then aggregates them to the target modelling scale, shifting the scale transformation from environmental inputs to parameters across different resolutions. This provides the benefit of minimizing uncertainties associated with environmental data aggregation and improving the representation of subgrid parameter heterogeneity. As evidenced by Samaniego et al. (2010), MPR exhibited superior performance to SR in maintaining water balance and preserving the spatial patterns of simulated variables, whereas the latter introduced larger deviations during the upscaling process. Given these advantages, MPR was selected as the default spatial parameter estimation approach in EVB. Beyond this implementation, we envision that future developments of EVB should further investigate scale-aware parameterization by incorporating scaling factor into transfer functions, allowing nonlinear upscaling relationships to be integrated into the spatial parameter estimation procedure. In our view, advancing spatial

parameter estimation approaches in this direction represents one of the key priorities for hydrological model development over the coming decade.

RC2-6/ 5. Computational efficiency is reported for only one JRB configuration; multi-resolution comparisons would be informative.

R/ Following your suggestion, we **now report runtime measurements for the GRB setup at three spatial resolutions**, replacing the original single-resolution results for the JRB setup. These results are presented at the end of Sect. 5.1 and in Table 3, providing a more informative assessment of EVB’s computational efficiency across modelling resolutions. The revised text is reproduced below for your review.

5.1 VIC Model deployment

To illustrate the computational efficiency of EVB, we used the built-in Logger class to record the average time required for each deployment step across the three spatial resolutions considered in the first case application (i.e., GRB), as described later; the results are reported in Table 3. The Logger class automatically records the execution time of each step and generates messages at predefined logging levels, providing traceability throughout the model deployment process. Building dpc instances (57% of the total runtime) and processing meteorological forcing data (41%) are the primary time-consuming steps, but both need to be executed only once for each modelling application and can thereafter be saved to persistent storage for convenient reuse. Because these operations are performed on a grid-cell basis, their computational costs increase approximately linearly with the number of grid cells as the spatial resolution becomes finer.

Table 3. Time consumption for EVB deployment workflow.

Step No.	Deployment Step	Description	Time Spent (seconds)		
			12 km	8 km	6km
1	Build dpc instances	Initialize data processing classes for handling input data	3251	7230	12700
2	Build domain file	Generate domain file	9	22	36
3	Build VIC parameter files	Generate parameter files at level-0 and level-1 scales	5	11	20
4	Build meteorological forcing files	Process and format meteorological data inputs	2315	5190	9200
5	Build RVIC parameter files	Prepare parameter files required by the RVIC routing model	18	39	71
6	Build global parameter file	Generate global parameters to control the VIC model run	1	1	1
7	Run VIC-RVIC coupled model	Perform one run of the VIC-RVIC coupled model	8	16	28
8	Sample and regenerate parameter files	Perform one round of parameter sampling and regeneration during calibration	5	8	12

RC2-7/ 6. A brief discussion of EVB vs. SUMMA+mizuRoute positioning would help readers understand use cases.

R/ Thank you for this helpful suggestion. We agree that a brief comparison with the SUMMA–mizuRoute framework would help clarify EVB’s positioning and scopes. We have therefore added this discussion to the **Introduction** and the **new Discussion section (Sect. 6.2)**. The comparison highlights the main advantages and intended scope of EVB. Further details are provided in our response to **RC2-2**.

Reference

- Gao, H., Tang, Q., Shi, J. X., Zhu, C., Bohn, T., Su, F., Sheffield, J., Pan, M., Lettenmaier, D., and Wood, E.: Water Budget Record from Variable Infiltration Capacity (VIC) Model, 2010.
- Gou, J., Miao, C., Duan, Q., Tang, Q., di, Z., Liao, W., Wu, J., and Zhou, R.: Sensitivity analysis - based automatic parameter calibration of the variable infiltration capacity (VIC) model for streamflow simulations over China, *Water Resources Research*, 56, 10.1029/2019WR025968, 2020.
- Gu, H. T., Liu, L., Bai, Z. X., Pan, S. L., and Xu, Y. P.: A stepwise surrogate model for parameter calibration of the Variable Infiltration Capacity model: the case of the upper Brahmaputra, Tibet Plateau, *J Hydroinform*, 23, 171-191, 10.2166/hydro.2020.010, 2021.
- Knoben, W., Clark, M., Bales, J., Bennett, A., Gharari, S., Marsh, C., Nijssen, B., Pietroniro, A., Spiteri, R., Tang, G., Tarboton, D., and Wood, A.: Community Workflows to Advance Reproducibility in Hydrologic Modeling: Separating Model-Agnostic and Model-Specific Configuration Steps in Applications of Large-Domain Hydrologic Models, *Water Resources Research*, 58, 10.1029/2021WR031753, 2022.
- Liang, X., Lettenmaier, D., Wood, E., and Burges, S.: A Simple hydrologically Based Model of Land Surface Water and Energy Fluxes for GSMs, *J. Geophys. Res.*, 99, 10.1029/94JD00483, 1994.
- Melsen, L., Teuling, A., Torfs, P., Zappa, M., Mizukami, N., Clark, M., and Uijlenhoet, R.: Representation of spatial and temporal variability in large-domain hydrological models: case study for a mesoscale pre-Alpine basin, *Hydrology and Earth System Sciences*, 20, 2207-2226, 10.5194/hess-20-2207-2016, 2016.
- Zheng, X., Liu, D., Wang, H., Ma, C., Liu, H., Ming, G., Li, Q., Khan, M. Y. A., and Hussain, F.: What can hydrological modelling gain from spatially explicit parameterization and multi-gauge calibration?, *Hydrol. Earth Syst. Sci.*, 30, 2493-2521, 10.5194/hess-30-2493-2026, 2026.

Finally, we would like to once again thank the Editor and all the Reviewers for your thorough review and support of our paper. If you have any questions, suggestions, or discussions, please feel free to contact us.