frontiers | Frontiers in Earth Science

TYPE Original Research
PUBLISHED 24 November 2022
DOI 10.3389/feart.2022.1026479

OPEN ACCESS

EDITED BY
Jeremy White,
Intera, Inc., United States

REVIEWED BY
Guoqiang Tang,
University of Saskatchewan, Canada
Zhongfan Zhu,
Beijing Normal University, China
Ayman Alzraiee,
California Water Science Center (USGS), United States

*CORRESPONDENCE
Maruti K. Mudunuru,
maruti@pnnl.gov

SPECIALTY SECTION
This article was submitted to Hydrosphere, a section of the journal Frontiers in Earth Science

RECEIVED 23 August 2022
ACCEPTED 03 November 2022
PUBLISHED 24 November 2022

CITATION
Mudunuru MK, Son K, Jiang P, Hammond G and Chen X (2022), Scalable deep learning for watershed model calibration.
Front. Earth Sci. 10:1026479.
doi: 10.3389/feart.2022.1026479

Scalable deep learning for watershed model calibration

Maruti K. Mudunuru*, Kyongho Son, Peishi Jiang, Glenn Hammond and Xingyuan Chen

Pacific Northwest National Laboratory, Richland, WA, United States

Watershed models such as the Soil and Water Assessment Tool (SWAT) consist of high-dimensional physical and empirical parameters. These parameters often need to be estimated/calibrated through inverse modeling to produce reliable predictions on hydrological fluxes and states. Existing parameter estimation methods can be time consuming, inefficient, and computationally expensive for high-dimensional problems. In this paper, we present an accurate and robust method to calibrate the SWAT model (i.e., 20 parameters) using scalable deep learning (DL). We developed inverse models based on convolutional neural networks (CNN) to assimilate observed streamflow data and estimate the SWAT model parameters. Scalable hyperparameter tuning is performed using high-performance computing resources to identify the top 50 optimal neural network architectures. We used ensemble SWAT simulations to train, validate, and test the CNN models. We estimated the parameters of the SWAT model using observed streamflow data and assessed the impact of measurement errors on SWAT model calibration. We tested and validated the proposed scalable DL methodology on the American River Watershed, located in the Pacific Northwest-based Yakima River basin. Our results show that the CNN-based calibration is better than two popular parameter estimation methods (i.e., the generalized likelihood uncertainty estimation [GLUE] and the dynamically dimensioned search [DDS], which is a global optimization algorithm). For the set of parameters that are sensitive to the observations, our proposed method yields narrower ranges than the GLUE method but broader ranges than values produced using the DDS method within the sampling range even under high relative observational errors. The SWAT model calibration performance using the CNNs, GLUE, and DDS methods are compared using R^2 and a set of efficiency metrics, including Nash-Sutcliffe, logarithmic Nash-Sutcliffe, Kling-Gupta, modified Kling-Gupta, and non-parametric Kling-Gupta scores, computed on the observed and simulated watershed responses. The best CNN-based calibrated set has scores of 0.71, 0.75, 0.85, 0.85, 0.86, and 0.91. The best DDS-based calibrated set has scores of 0.62, 0.69, 0.8, 0.77, 0.79, and 0.82. The best GLUE-based calibrated set has scores of 0.56, 0.58, 0.71, 0.7, 0.71, and 0.8. The scores above show that the CNN-based calibration leads to more accurate low and high streamflow predictions than the GLUE and DDS sets. Our research demonstrates that the proposed method has high potential to improve our current practice in calibrating large-scale integrated hydrologic models.

KEYWORDS

SWAT calibration, watershed modeling, parameter estimation, inverse problems, convolutional neural networks, scalable deep learning

1 Highlights

- We developed a scalable deep learning (DL) methodology to estimate SWAT model parameters.
- Our DL methodology is based on convolutional neural networks (CNN).
- Our CNN-enabled SWAT model calibration shows higher streamflow prediction accuracy than traditional parameter estimation methods such as the Generalized Likelihood Uncertainty Estimation (GLUE) and the Dynamically Dimensioned Search (DDS) algorithms.
- Estimated SWAT model parameters from observed discharges are within the sampling range of ensemble simulations even when high-observational errors exist.
- An added benefit is that CNN-enabled parameter estimation after training is at least $\mathcal{O}(10^3)$ times faster than GLUE- and DDS-based methods.
- However, the hyperparameter tuning to discover reasonably accurate CNN models is computationally expensive, which is in $\mathcal{O}(10^5)$ processor hours.

2 Introduction

Watershed models frequently are used to predict streamflow and other components in the terrestrial water cycle. These components are affected by a wide range of anthropogenic activities (e.g., agricultural intensification), climate perturbations (e.g., rain-on-snow, rising temperatures and increasing precipitation, earlier occurrence of snow melt in mountainous regions), and disturbances (e.g., wildfire) (Singh and Frevert, 2003; Singh and Frevert, 2010; Daniel et al., 2011). Watershed models also have been used to assess the sustainability of the water supply for effective water resource management. Some popular and open-source watershed modeling software that can accurately simulate various components of water cycling in intensively managed watersheds include the Soil and Water Assessment Tool (SWAT) and its variants (e.g., SWAT-MRMT-R) (Mankin et al., 2010; Neitsch et al., 2011; Fang et al., 2020), the Advanced Terrestrial Simulator (ATS) (Coon et al., 2020), the Precipitation Runoff Modeling System (PRMS) (Leavesley et al., 1983; Markstrom et al., 2015), the Weather Research and Forecasting Model Hydrological modeling system (WRF-Hydro) (Sampson and Gochis, 2018; Wu et al., 2021), etc (Donigian et al., 1995; Tague and Band, 2004; Graham and Butts, 2005; Cuo et al., 2008; Hamman et al., 2018).

Watershed models adopt physical laws (e.g., mass and energy balance) or known empirical relationships to simulate the watersheds' different hydrological components (e.g., infiltration, evapotranspiration, groundwater flow, streamflow). These models feature two types of parameters (Johnston and Pilgrim, 1976; Mein and Brown, 1978; Nakshatrala and Joshaghani, 2019). The first type includes parameters with physical characteristics (e.g., permeability, porosity). The second type includes conceptual or empirical parameters, which are currently impossible or difficult to measure directly. Most watershed simulators (e.g., SWAT, PRMS) consist of parameters that fall in the second category (Singh and Frevert, 2010). As a result, observed data, such as streamflow collected at the watershed outlet, are used to estimate the conceptual parameters through model calibration. Many semi-distributed or bucket models can only achieve adequately accurate predictions after calibrating their parameters with available observations, making them less ideal for ungauged watersheds. On the other hand, advanced fully integrated watershed models (e.g., ATS) can predict watershed responses with reasonable accuracy without undergoing intensive model calibration; however, running those models is computationally expensive (Chen et al., 2021; Cromwell et al., 2021). Certain parameters in these mechanistic models (e.g., ATS) are measurable and physically significant while others are empirically similar to the SWAT.

Various techniques and software tools for calibrating watershed models have been reported in the literature (Duan et al., 2004). Popular methods include generalized likelihood uncertainty estimation (GLUE) (Blasone et al., 2008; Nott et al., 2012), the dynamically dimensioned search (DDS), maximum likelihood estimation (Myung, 2003), the shuffled complex evolution method developed at the University of Arizona (SCE-UA) (Duan et al., 1994), Bayesian parameter estimation methods (Thiemann et al., 2001; Gupta et al., 2003; Misirli et al., 2003), ensemble-based data assimilation methods (e.g., ensemble Kalman filter, ensemble smoother) (Evensen, 1994; Van Leeuwen and Evensen, 1996; Evensen, 2003; Chen et al., 2013; Evensen, 2018; Jiang et al., 2021), and adjoint-based methods (Tarantola, 2005; Aster et al., 2018). These techniques underpin popular software packages such as PEST (Doherty and Hunt, 2010), DAKOTA (Adams et al., 2009), SWAT-CUP (Abbaspour, 2013), MATK (Model Analysis ToolKit, 2021), MADS (MADS, 2021), and DART (Anderson et al., 2009), which are developed to facilitate model calibration. Using these existing calibration methods and tools can be time consuming (e.g., slow convergence), require good initial guesses, and can be computationally intensive (e.g., may require many forward model runs or runs using high-performance computing clusters) (Rouholahnejad et al., 2012; Zhang et al., 2016; Bacu et al., 2017). Moreover, calibration using such tools can potentially result in reduced accuracy when estimating high-

dimensional parameters (> 10) (Duan et al., 2004; Eckhardt et al., 2005). New PEST tools have been developed to handle high dimensional inverse modeling like PESTPP-ies and PESTPP-DA. However, many of the methods mentioned above have challenges (see Supplementary Text S1) in properly capturing the strong nonlinear relationships between parameters and observed responses (Franco and Bonumá, 2017). Recent advances in deep learning (DL) (e.g., deep neural networks [DNNs], convolutional neural networks [CNNs]) show promise for developing reliable model calibration methods that overcome the challenges described above (Gabrielli et al., 2017; Cromwell et al., 2021).

Deep learning shows promise in aiding inverse modeling associated with highly nonlinear relationships (Zhang et al., 2009; Gabrielli et al., 2017; Marçais and de Dreuzy, 2017; Afzaal et al., 2020; Sit et al., 2020; Nearing et al., 2021). It uses multiple neural layers to extract features that are representative of inputs, and DL-enabled inverse models for parameter estimation are known to be robust even when observed errors or noise exist (Rolnick et al., 2017; Edwards, 2018; Gupta and Gupta, 2019; Rudi et al., 2020). In hydrology, neural networks (e.g., deep, convolutional, recurrent) have been used to model and predict streamflow, water quality, and precipitation (Shen, 2018; Khandelwal et al., 2020; Bhasme et al., 2021). Recently Tsai and co-workers (Tsai et al., 2021) developed a novel differentiable parameter learning framework that efficiently learns a global mapping between inputs and process model parameters. They applied this framework to estimate Variable Infiltration Capacity (VIC) land surface hydrologic model. The trained DL models produced parameters which allow VIC to best match surface soil moisture observations from NASA's Soil Moisture Active Passive satellite mission. In this paper, we present a scalable, DL methodology that uses observed streamflow data to estimate high-dimensional SWAT model parameters efficiently and reliably with reasonable accuracy. By scalable, we mean that the CNNs can be trained and tuned at any scale (e.g., from laptop computers to high-performance computers at leadership-class computing facilities) without any changes in the proposed method or developed code. This study uses CNNs, which are frequently used in hydrological applications (Sadeghi et al., 2019; Van et al., 2020; Jagtap et al., 2021).

CNNs offer many advantages over DNNs (Read et al., 2019; Dagon et al., 2020; Jia et al., 2021; Rahmani et al., 2021; Willard et al., 2022). A significant advantage of CNNs is that they explicitly learn local representations (or patterns). As a result, CNNs are best suited to produce image or time series data where the neighboring dependencies are important. This superior performance of CNNs can be attributed to the multiple convolutional layers that learn hierarchical patterns from the inputs. The resulting broader set of abstract patterns are used to develop nonlinear mappings between streamflow and the SWAT model parameters. Another benefit of CNN-enabled inverse models is their low inference time for parameter estimation compared to traditional methods; however, data requirements and associated training time (e.g., hyperparameter tuning) needed to develop such inverse models can be substantial. Once the CNN-enabled inverse model is trained, it can allow assimilation of observed data, thereby significantly reducing the time required to estimate parameters in high-dimensional space (Cromwell et al., 2021).

2.1 Main contributions

The main contribution of this study is development of an accurate parameter estimation methodology using CNNs that calibrates watershed models better than traditional methods (e.g., GLUE, DDS). The CNN-enabled inverse mappings are built on ensemble simulations generated by the SWAT model. Scalable hyperparameter tuning is performed to identify the top 50 architectures based on mean squared error and other performance metrics[1]. Further, we test the influence of errors in observed streamflow on parameter estimation and streamflow prediction accuracy. A significant advantage of the proposed DL method is that it estimates sensitive parameters with reasonably good accuracy even at high observation error levels (e.g., 100% relative observational errors). Moreover, these estimated parameters are within the prior sampling range, showing the proposed methodology's robustness to observational errors. Compared to the GLUE and DDS optimization methods, parameters estimated by the CNN-enabled inverse model provide more accurate streamflow predictions within and beyond the calibration period. The GLUE method identified a set of behavioral parameters within the ensemble parameter combinations. By "behavioral" parameters, we mean to signify parameter sets for which SWAT model simulations are deemed to be "acceptable" upon satisfying certain user-defined performance metrics (e.g., KGE greater than 0.5) on observed data (Blasone et al., 2008). Based on a cutoff threshold that uses metrics such as the KGE, the entire set of simulations then is split into behavioral and non-behavioral parameter combinations. The behavioral parameter set provides better accurate predictions than the non-behavioral set. Our analysis also showed that the CNN estimated parameter sets are narrower than the GLUE-based behavioral sets but wider than estimations obtained using the DDS method. As the DDS method is a global optimization, it searches for a best parameter value based on a performance metric (e.g., KGE). Hence, the obtained parameter ranges from the DDS method can be narrower than those

1 Popular objective functions such as R^2-score, Nash-Sutcliffe efficiency (NSE), Kling-Gupta efficiency (KGE), and their modifications (e.g., logNSE, mKGE, and npKGE) are used to evaluate the fit between observed and simulated streamflow time series.

obtained from the CNN and GLUE methods. Another advantage of the proposed CNN-based inverse models is that it is at least $\mathcal{O}(10^3)$ times faster than the GLUE and DDS methods. From a computational cost perspective, traditional parameter estimation using local and global optimization algorithms (e.g., using PEST, DAKOTA) requires multiple forward model runs. As a result, inverse modeling may require source code modifications and also high-performance computational resources, which can be prohibitively expensive. We acknowledge that hyperparameter tuning can be expensive. However, such tuning is needed for finding optimal CNN architectures. Once CNNs are trained, the savings in computational cost enable our DL-enabled parameter estimation to be inclusive [i.e., easy to adapt using transfer learning (Zhuang et al., 2020; Song and Tartakovsky, 2021)] and ideal for calibrating multi-fidelity models (e.g., ATS, PFLOTRAN, WRF-Hydro, PRMS) at spatial scales of watersheds and basins.

2.2 Outline of the paper

The paper is organized as follows: Section 2 discusses state-of-the-art methods for parameter estimation and their limitations. We also demonstrate the need for developing DL method to better calibrate hydrological models, such as SWAT. Section 3 describes the study site and SWAT model developed using a National Hydrography Dataset PLUS (NHDPLUS v2)-based watershed delineation (Moore and Dewald, 2016). We discuss data generation to develop CNN-enabled inverse models. We also compared observed data with the SWAT model ensemble simulations. Section 4 introduces the proposed scalable DL methodology for estimating SWAT parameters. We performed sensitivity analysis to rank the sensitivities of SWAT model parameters. We performed scalable hyperparameter tuning to identify the optimal CNN architectures and described the associated computational costs for training the DL models and generating inferences (e.g., on test and observational data). Section 5 presents the training, validation, and testing results of the CNNs. We compared the performance of CNN-estimated parameters with those of the GLUE and DDS methods. Performance of the calibration model within and beyond calibration period is provided. Sections 6 and 7 present our future work and conclusion.

3 Study site and data generation

This section first describes the study site, the American River Watershed (ARW) in the Yakima River Basin (YRB), before discussing the SWAT model, its parameters, and specifics on ensemble runs needed to develop CNN-enabled inverse models. We also compare the observed streamflow data used to calibrate the SWAT model with the ensemble runs within the calibration period (i.e., from water years [WYs] 2014 to 2016 [1 October 2013, through 30 September 2016]). Each SWAT model run produces daily simulated streamflow values.

3.1 Study site

The YRB (see Figure 1), situated in Eastern Washington State, has a drainage area of about 16,057 km^2 (Mastin and Vaccaro, 2002; Qiu et al., 2019). The daily averaged flow for the YRB is about 95 m^3s^{-1} over a period of 40 years. This averaged flow is computed using the data collected from 1/1/1980 to 12/31/2021 at the Kiona gauge station, which is the closest to the outlet of the YRB. A major tributary of the Yakima River is the American River, which is a third-order stream, with a watershed of about 205 km^2. According to 30-year normalized PRISM data, the mean annual precipitation and temperature within the ARW range from 978 to 2,164 mm and 2.8–4.9°C, respectively (Daly et al., 2000; Daly and Bryant, 2013; PRISM, 2021). The climate within the ARW exhibits strong seasonal patterns, including cold, wet winters and hot, dry summers. About 60% of precipitation occurs in the winter as snow, with snowmelt occurring from April to June the following year. Peak snow accumulation and flow occur in April and May, respectively. This prior site-specific knowledge shows that the snow process parameters in the SWAT model are essential. Guided by the information mentioned above and sensitivity analysis, our results demonstrate that we can better estimate such important process model parameters using our DL method rather than DDS and GLUE methods.

The slope of the ARW varies from 0° to 83°, with a mean slope of 23°. The major surface geology types are andesitev (72%), granodiorite (20%), and alluvium (8%). The primary soil texture is gravelly loamy sand with a maximum soil depth of 1,524 mm based on U.S. Department of Agriculture State Soil Geographic Data (STATSGO) (Schwarz and Alexander, 1995). This soil is classified as hydrologic group B with moderate runoff potential and infiltration rates. Evergreen trees (83%) and shrub (11%) dominate the land cover. Other types of land cover include urban, grass, and wetlands. The ARW has a U.S. Geological Survey (USGS) gauging station (USGS 12488500) located in the watershed outlet. This station has been recording the daily observed streamflow from 16 July 1988, to the present. A snow telemetry (SNOTEL) station (site name: Morse lake) is located northwest of the watershed. This SNOTEL station has measured the snow water, daily precipitation, and air maximum/mean/minimum temperatures from 1 October1979, to the present.

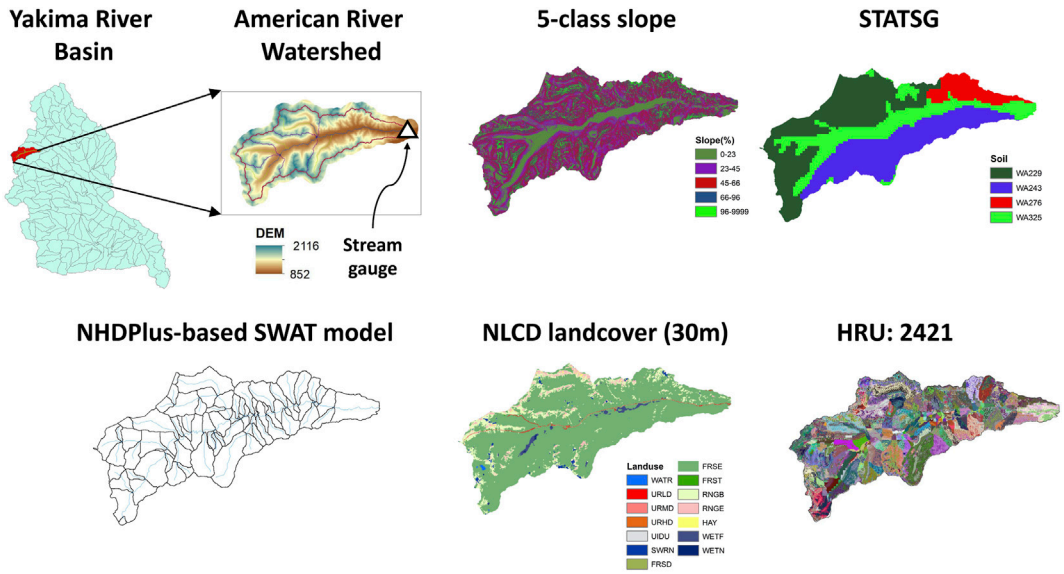

FIGURE 1
NHDPlus-based SWAT model for ARW: This figure shows the watershed delineation and data products used to develop the SWAT model for our study site. The top left figures show the delineation of the YRB into different watersheds, including our ARW study site. It also shows the DEM used for modeling the ARW and the associated stream gauge at the watershed outlet. The top right figures show the spatially varying slope and soil data within the study site. The bottom left figure shows the delineation of third-order streams in the ARW. The bottom right figures show the spatially varying land cover (primarily evergreen) and the number of HRU employed in the SWAT model.

3.2 Brief description of the SWAT model

SWAT is a semi-distributed eco-hydrological model. It can simulate both subsurface and surface hydrological processes, soil or plant bio-geochemistry, and in-stream processes (Arnold et al., 2012). The SWAT model requires various spatial Geographic Information System data to represent the different watershed characteristics (e.g., topography, land cover, and soil). The USGS 10-m digital elevation model (DEM) is used to compute the topographic parameters (e.g., drainage area, slope, slope length) with the ARW basin and sub-basin boundaries and stream networks defined by the National Hydrography Dataset Plus (NHDPlus) catchment/streams. Previous studies (Chiang and Yuan, 2015; Moore and Dewald, 2016) have demonstrated that NHDPlus-based catchment/streams outperformed modeled streamflows that did not account for such delineation.

Figure 1 shows the location of the ARW, and key inputs of NHDPlus-based SWAT model used to simulate streamflow at the ARW study site. This model is composed of 87 sub-basins with five slope classes (percent rise of the slope): 1) 0–26, 2) 26–51, 3) 51–80, 4) 80–129, and 5) 129–999. Data from the USGS National Landcover Database 2016 (30-m resolution) and the Department of Agriculture STATSGO database are used to estimate the land cover/use and soil parameters, respectively. Hydrologic response unit (HRU) maps are developed by fully combining unique slope class, land cover/use, and soil type, resulting in a total of 2,421 HRUs for the ARW (see Figure 1). Supplementary Text S1 provides additional details on SWAT model development for our study site. Daily precipitation, maximum and minimum air temperatures, radiation, and relative humidity from a daily Daymet (Daymet, 2021) with 1-km spatial resolution are used to prepare the climate input data for the SWAT model simulations. Wind speed data are generated using weather generators in the SWAT.

3.3 Data for the SWAT model calibration

Table 1 summarizes the 20 parameters and their associated sample ranges (e.g., minimum and maximum values) we calibrated in the SWAT model to generate simulation data. The table clearly shows seven groups/types of SWAT model parameters: 1) landscape, 2) soil, 3) groundwater, 4) channel, 5) snow, 6) plant, and 7) climate. Each parameter in a specified group is calibrated at different spatial scales. For example, snow group parameters such as SFTMP and SMTMP represent basin-scale snow processes. Channel group parameters are at the sub-basin level, and soil/groundwater/plant/climate group parameters represent HRU level spatial variation. Even though some parameters differ at the HRU level, we calibrate basin-scale scaling coefficients that vary within [−0.3, 0.3] and are the same for all HRUs.

TABLE 1 This table provides a list of 20 different SWAT model parameters that are calibrated using the proposed scalable DL methodology. The associated lower and upper limits of parameter values also are specified. Boldfaced descriptors are mutual information (MI)-identified sensitive parameters (Jiang et al., 2022b).

Parameter group/type	Parameter[2]	Lower limit	Upper limit	Brief description (units)	Parameter modification[3]	Spatial variability
Landscape	**CN2**	−0.3	0.3	% change in SCS runoff curve number	R	Varying across HRUs
Groundwater	**RCHRG_DP**	0	1	Deep aquifer percolation fraction	V	Constant across HRUs
Groundwater	GWQMN	0	5,000	Threshold depth of water in the shallow aquifer required for return flow to occur (mm)	V	Constant across HRUs
Groundwater	GW_REVAP	0	0.2	Groundwater "revap" coefficient	V	Constant across HRUs
Groundwater	REVAPMN	1	500	Threshold depth of water in the shallow aquifer for "revap" to occur (mm)	V	Constant across HRUs
Groundwater	**GW_DELAY**	1	100	Groundwater delay (days)	V	Constant across HRUs
Groundwater	**ALPHA_BF**	0.01	0.99	Baseflow alpha factor	V	Constant across HRUs
Soil	SOL_K	−0.3	0.3	% change in saturated hydraulic conductivity (mm h^{-1})	R	Varying across HRUs
Soil	SOL_AWC	−0.3	0.3	% change in available water change in capacity of the soil layer (mm H_2O mm $soil^{-1}$)	R	Varying across HRUs
Soil	ESCO	0.01	1	Soil evaporation compensation factor	V	Constant across HRUs
Soil	OV_N	−0.3	0.3	% change in Manning's "n" value for overland flow	R	Varying across HRUs
Channel	**CH_K2**	0	200	Effective hydraulic conductivity in main channel alluvium (mm h^{-1})	V	Constant across sub-basins
Channel	**CH_N2**	0.02	0.15	Manning's "n" value for the main channel	V	Constant across sub-basins
Snow	**SFTMP**	−5	5	Snowfall temperature (°C)	V	Constant in the basin
Snow	**SMTMP**	−5	5	Snow melt base temperature (°C)	V	Constant in the basin
Snow	**SMFMX**	1.4	6.9	Maximum melt rate for snow during the year (mm H_2O°C day^{-1})	V	Constant in the basin
Snow	**TIMP**	0.01	1	Snowpack temperature lag factor	V	Constant in the basin
Plant	**EPCO**	0.01	1	Plant uptake compensation factor	V	Constant in the basin
Climate	PLAPS	343.3	964	Precipitation lapse rate (mm km^{-1})	V	Constant in the basin
Climate	TLAPS	-4.86	3.353	Temperature lapse rate (°C km^{-1})	V	Constant in the basin

Using a Sobol quasi-random[2] sequence sampling method (Herman and Usher, 2017), we generated 1,000 sets of these 20 parameters to develop CNN-enabled inverse models. Sobol sequence is quasi-random low-discrepancy sequences (Sobol', 1967; Herman and Usher, 2017). Compared with random sampling from a uniform distribution, Sobol sequence guarantee better uniform coverage of the samples. We adopted Sobol sequences to generate the ensemble realizations of standardized parameters within [0,1], which were then scaled back to the parameter ranges shown in Table 1 The daily streamflow data and flow duration curves simulated using the SWAT model for these 1,000 realizations are shown in Figure 2. The simulation time for the SWAT model calibration is between the beginning of WY 2014 to the end of WY 2016 (i.e., 1 October 2013—30 September 2016), which is referred to as the calibration

2 Table 1: Note that the sensitive parameters are identified using the MI method. These sensitive parameters are presented in boldface in this parameter column.

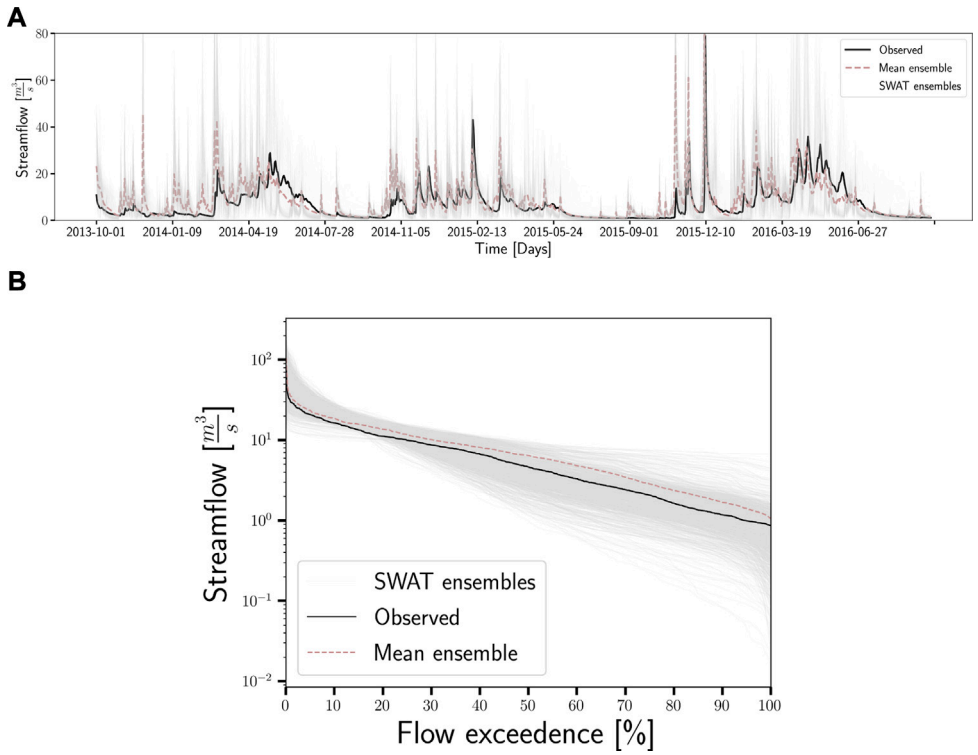

FIGURE 2
SWAT model simulations vs. observational data (within calibration period): This figure compares the modeled streamflow data generated based on NHDPlus-based SWAT model with observed flow for the ARW study site. The top figure **(A)** shows the streamflow time-series and the bottom figure **(B)** shows the flow duration curves. The dark brown color dashed line represents the ensemble mean of 1000 SWAT model simulations. The grey color lines represents the modeled streamflow ensembles. The black-colored line corresponds to the observed streamflow data.

period. The validation period is from[3] WY 2000 through WY 2013 (i.e., 1 October 1999—30 September 2013). The calibrated SWAT model is run during the validation period, and its performance is then compared with the observed data. Figure 2 compares the ensemble mean of simulated discharge (i.e., 1,000 realizations) with the observational data. The grey color represents each of the 1,000 simulated discharge realizations. This streamflow time-series and flow duration curve qualitatively shows the similarities of the trends in the simulated discharge and observed data. However, the comparison against observations also show the over/under predictions of peak/low flows that can be due to structural deficiencies of the model. The SWAT model fidelity may need to be enhanced to overcome these structural deficiencies. The generated data are used to estimate SWAT parameters by the CNN-based calibration, GLUE, and DDS methods. The GLUE- and DDS-based SWAT model calibrations also are compared with the observed data for both periods. The behavioral model parameter sets (i.e., from the GLUE method) are selected based on KGE metrics. We also use the other accuracy measures (e.g., NSE, logNSE, R^2-score) to evaluate the calibrated SWAT model performance, which are described in Section 4.4.

4 Proposed methodology

This section presents the overall methodology consisting of data pre-processing, scalable hyperparameter tuning (Mudunuru et al., 2022), and computational cost of constructing the CNN-enabled inverse models. We also briefly describe the GLUE and DDS optimization methods that are used to compare the performance of CNNs. The comparison of the DL method performance against the most commonly applied algorithms for calibration of

3 In Table 1, the parameter modification column indicates how SWAT model parameters are modified during calibration and the training data generation for CNN-enabled inverse modeling. The term "V" indicates that existing SWAT model parameter values are replaced with values in the provided range. The term "R" indicates relative changes in parameters by multiplying existing values with 1+ calibrated parameter values in the range (Qiu et al., 2019). The CN2, SOL_K, and SOL_AWC parameter modifications are "R," whose absolute values as (Eckhardt et al., 2005; Rouholahnejad et al., 2012), [0.001, 1,000], and [0.01, 0.35], respectively.

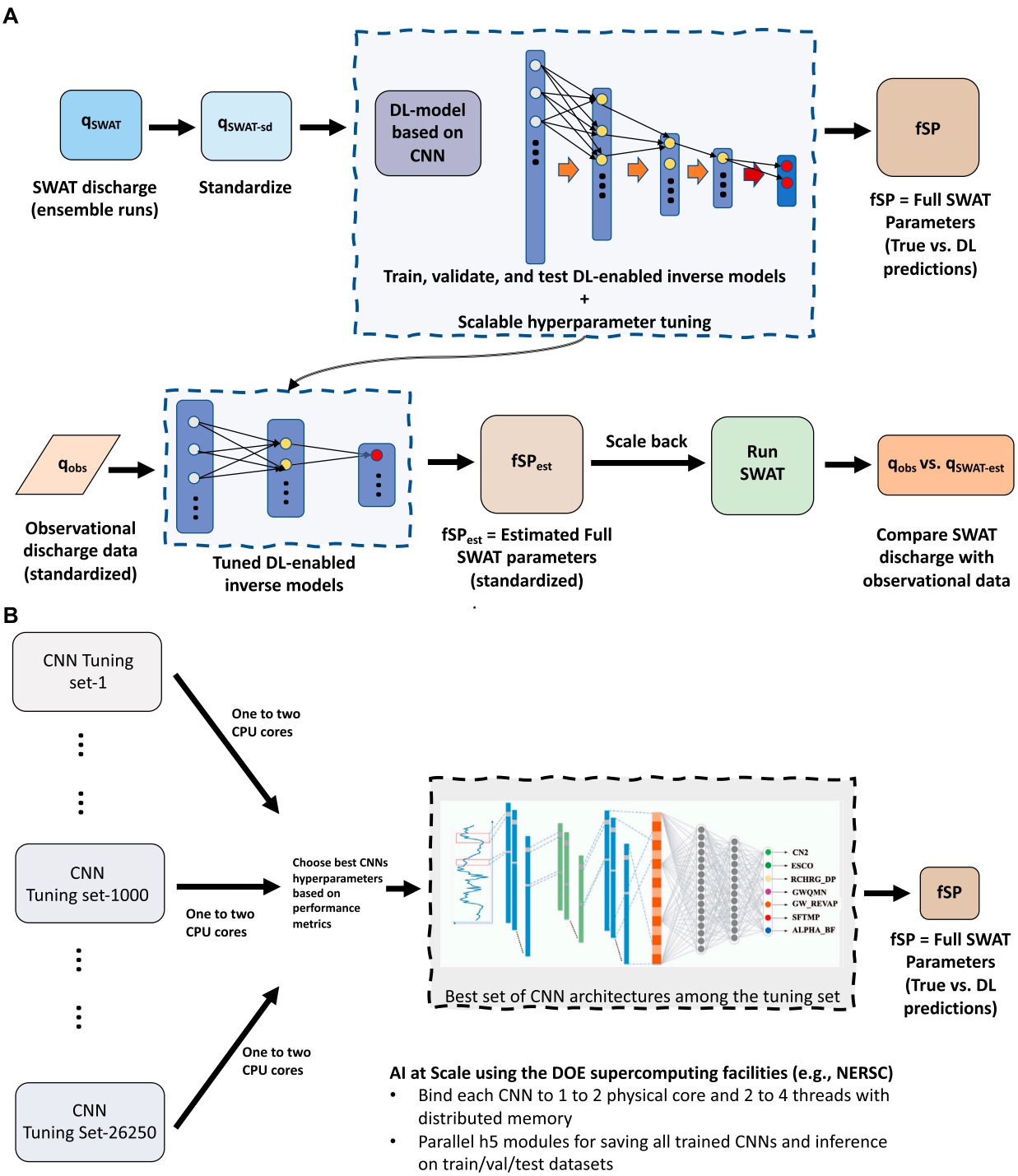

FIGURE 3
Proposed scalable deep learning workflow for the SWAT model calibration: A pictorial description of the proposed DL methodology to estimate parameters and calibrate the SWAT model using observational discharge. Ensemble simulations generated by the SWAT model are used to train, validate, and test the CNN-enabled inverse models, as shown in the top figure **(A)**. The observed streamflow is then provided as an input to the developed DL models to estimate site-specific parameters. These parameters are then used by the SWAT model to simulate discharge for comparison with observational data. The bottom figure **(B)** shows a scalable hyperparameter tuning approach to identify optimal CNN architectures using high-performance computing resources at NERSC. Each explored CNN architecture is trained on one/two CPU physical cores and its performance is estimated using validation loss and streamflow prediction metrics (e.g., R^2-score, NSE, logNSE, KGE and its variants). From the explored space, top 50 CNNs are chosen for inverse modeling and analysis.

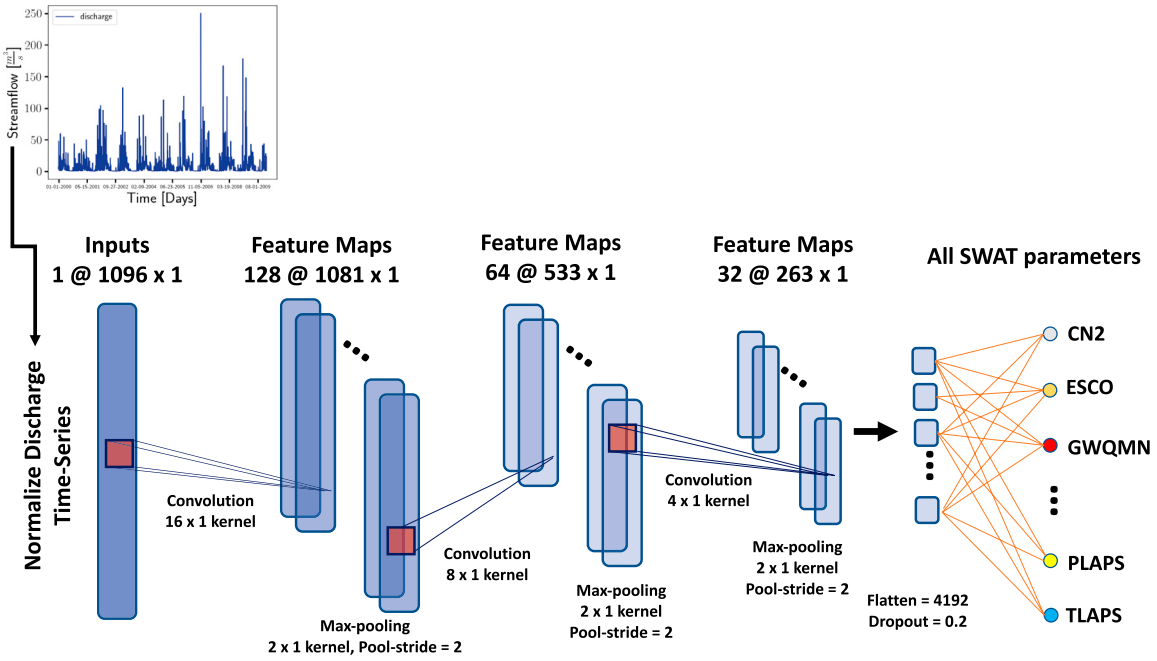

FIGURE 4
A top-50 CNN-enabled inverse model architecture: This figure shows a pictorial description of a deep CNN for estimating SWAT model parameters. This CNN model inversely maps simulated discharge to 20 different conceptual parameters. Hyperparameter tuning is then performed to arrive at this CNN architecture. This CNN-enabled inverse model is a multi-task learning model that accounts for the correlation between different SWAT parameters during training.

watershed simulation models (i.e., DDS and GLUE) gives better insight into CNN's capability in providing accurate parameter estimations and uncertainty on streamflow predictions.

4.1 Proposed scalable deep learning methodology

Figure 3 summarizes our proposed DL method for training the inverse models and then inferring the SWAT parameters. We train, validate, and test CNN-enabled inverse models (Schmidhuber, 2015; Goodfellow et al., 2016; Chollet, 2017) using SWAT model ensemble runs. The proposed DL methodology can be divided into multiple steps, which is described below in a step-by-step approach.

1) The inputs to the CNNs are the modeled daily streamflow time-series data and outputs are the SWAT parameters. Both the inputs and outputs are normalized for training CNNs.
2) The CNN-enabled inverse models are developed to estimate all 20 of the SWAT process model parameters. We used the Keras API in Tensorflow package (Keras API, 2021) to build our CNN-enabled inverse models.
3) The simulated streamflow and parameter sets are assembled into a data matrix and then partitioned into training (80%), validation (10%), and testing (10%) sets, of which each SWAT run contains 1,096 daily data points. The training and validation sets are jointly used in hyperparameter tuning to find the optimal CNN architecture.
4) The dataset is normalized, which is necessary for CNN model development as CNNs are filter/kernel-based methods that benefit from normalization of their inputs to make accurate predictions (Anysz et al., 2016; Gu et al., 2018). The normalization is done by first removing the mean and scaling the training dataset to unit variance and then applying the same pre-processing normalizer to transform the validation and testing sets.
5) Hyperparameter tuning is performed to identify the optimal CNN architectures whose performances were evaluated against validation dataset during model training. By CNN architecture, we mean convolutional and pooling layers that needs to be tuned for optimal performance.
6) The testing step includes performance evaluation (e.g., mean-squared error) of the tuned CNNs on test data.
7) The observed data are standardized using the pre-processing normalizer (Pedregosa et al., 2011) that we trained on simulation data. This normalized data is input to the tuned

TABLE 2 This table provides the hyperparameter space used to explore CNN architectures for developing reliable DL-enabled inverse mappings for the SWAT model calibration.

Hyperparameter type	Description	Explored options
Layers	Number of 1D convolutional layers	[1, 2, 3, 4, 5]
Filters	The number of output filters in the 1D convolution	[16, 32, 64, 128, 256]
Kernel size	An integer to specify the length of the 1D convolution window	[2, 4, 8, 16, 32]
Dropout rate	Applies dropout to the input[5]	[0.0, 0.1, 0.2, 0.3, 0.4]
Learning rate	The value of the optimizer in the Adam algorithm	[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]
Batch size	The number of training samples seen by CNN per gradient update	[4, 8, 16, 32, 64]
Epochs	The number of times the algorithm sees the training data	[50, 100, 200, 300, 400, 500]

CNN-enabled inverse model to estimate the study site SWAT model parameters. We also add errors to observed streamflow data and assess the performance of CNNs for SWAT model calibration.

8) Finally, these calibrated parameter sets are given to the SWAT model to obtain daily streamflow values in the calibration and validation periods. The predicted discharge is then compared with the observed data to evaluate the performance of the CNN-calibrated SWAT model in both calibration (WY 2014–2016) and validation time periods (WY 2000–2013).

Hyperparameter tuning is a crucial step in obtaining reliable and accurate CNN-enabled inverse models. The search for hyperparameters is performed in parallel at the National Energy Research Scientific Center (NERSC) (NERSC, 2021), a high performance computing user facility operated by Lawrence Berkeley National Laboratory for the U.S. Department of Energy Office of Science. Scalable hyperparameter tuning is achieved by combining mpi4py (MPI for Python) package with Tensorflow package and parallel HDF5 modules to train each CNN architecture on at least one physical central processing unit (CPU) (see Figure 3B). As tuning is embarrassingly parallel, the CNN architectural search space is distributed across the processes employed and run simultaneously on one to two cores each. All trained CNN models and their inferences are written to their individual HDF5 files. This tuning is necessary as the training process and predictions of the CNN-enabled inverse model are controlled by the parameters and topology of the CNN architecture. We tested two types of hyperparameters: 1) model hyperparameters and 2) algorithm hyperparameters. Model hyperparameters define the neural network architecture. For instance, the selection of CNN topology is influenced by model hyperparameters such as the number and width of hidden layers. Algorithm hyperparameters influence the training process after the architecture is established. The values of the trainable weights of a CNN architecture are controlled by algorithm hyperparameters such as learning rate and the number of epochs. Table 2 shows the search space that we explored. Supplementary Table S1 shows the model and algorithm hyperparameters for the top 50 CNN architectures identified through this scalable approach. During the tuning process, we used ReLU as the activation function, and max pooling is taken to be equal to 2. The optimal hyperparameter set is chosen based on the validation mean squared error along with streamflow prediction metrics using the grid search tuning method[4]. In addition to identifying the optimal hyperparameter set, we also identified the next 50 best candidates. In Section 5, we show the predictions of these 50 best models and the associated uncertainty in their streamflow predictions[5].

Figure 4 shows a pictorial description of a tuned CNN architecture from the grid search. The CNN filters are initialized with the Glorot uniform initializer (Gu et al., 2018; Keras API, 2021). This Glorot uniform allows us to initialize the weights so the variance of the activations are the same across every neural layer. Moreover, this constant variance initialization helps prevent the gradient from exploding or vanishing. After each convolution, a max-pooling operation is applied, and the final convolutional layer is flattened. After the dropout layer, the remaining features are mapped to the SWAT parameters. The entire CNN is compiled using an Adam optimizer, with the loss being the mean squared error. The resulting tuned CNN architectures (each of the top 50 models) have approximately 1 M trainable weights.

4.2 Dynamically dimensioned search method

The DDS method is a global optimization algorithm developed to automatically calibrate highly parameterized

4 Grid search is an exhaustive search technique performed on a specific hyperparameter values of the CNN architecture.

5 Table 2: To reduce model over-fitting, we randomly set the last convolutional layer units that connect to the output to 0 at each step during training time. The rate value controls the frequency of dropping the units.

hydrologic models. Typically, the total number of evaluations available for SWAT model calibration is always limited and is also case-study dependent because of the curse of dimensionality. The DDS method is designed from this calibration perspective to find practical or high-quality parameter sets. It is well known that the DDS method outperforms methods such as SCE-UA (available in PEST package) when the number of calibrated parameters is high (i.e., 10 or more) (Tolson and Shoemaker, 2007). Below, we summarize the steps involved in executing the DDS method to calibrate the SWAT model.

First, we define DDS algorithm inputs such as neighborhood perturbation size parameter (0.2 as the default value), maximum number of function evaluations (a total of 500 for each random seed), number of random seeds (a total of 10), bounds on all the SWAT model parameters (as mentioned in Table 1), and initial guesses/solutions for these parameters. Second, for the initial guess, we construct and evaluate an objective function (e.g., KGE) that minimizes differences between the simulated and observed data. Third, we perturb the initial guess by using a vector sampled from a standard normal random distribution with zero mean and unit standard deviation. We ensure that the perturbed values are within the physical bounds, which is the SWAT parameter range. Fourth, we evaluate the objective function and update the best solution until all user-defined evaluations are exhausted or a stopping criterion is met. We executed these steps for 10 different random seeds, which resulted in a total of 5,000 DDS calibration sets (i.e., 10×500). Then, we selected the top-50 from this total of 5,000 DDS calibration sets.

4.3 Generalized likelihood uncertainty estimation method

The GLUE method (Beven and Binley, 2014) used in hydrology provides a framework for evaluating model performance and quantifying the impact of various uncertainty sources on predictive uncertainty. For its simplicity and flexibility, the GLUE method (Beven and Binley, 2014) has been applied to various watershed models. The method uses a Monte Carlo approach to evaluate different model structure/parameter sets by comparing observed data with modeled values. In many cases, the different models or parameter sets show similar model performance (e.g., NSE), which is called as an equifinality. Thus, instead of searching for an optimum model, searching for a behavioral parameter and model structure is a general practice. In this study, we use the GLUE method to select the behavioral parameter sets for the SWAT model by comparing the observed streamflow and modeled value. Because of the lack of prior knowledge of the distribution of each parameter, the 20 parameters used in the SWAT model are assumed to follow uniform distributions, and we use a Sobol sequence method to efficiently sample the parameters values. The behavioral parameter sets are the top-50 sets selected from a total of 1,000 simulations based on the accuracy of the KGE metric. The selected KGE values of the behavioral parameter sets range from 0.5 to 0.7. They are shown in Section 5 and also in Supplementary Table S2. Also, to evaluate the impact of total number of model simulations on model performance, we also increased the number of model simulations from 1,000 to 5,000, and the results obtained from 5,000 simulations remain very similar to the results from 1,000 simulations.

4.4 Performance metrics

The evaluation criteria for SWAT model calibration using the CNN, DDS, and GLUE estimated sets include R^2-score, NSE, logNSE, KGE, and its variants (i.e., mKGE and npKGE) (Hydroeval, 2021). For instance, NSE, logNSE, and KGE are evaluated as follows:

$$\mathrm{NSE}(\mathbf{q}, \hat{\mathbf{q}}) = 1 - \frac{\sum_{i=1}^{n} (q_i - \hat{q}_i)^2}{\sum_{i=1}^{n} (q_i - \mu_{\mathbf{q}})^2} \quad \text{where} \quad \mu_{\mathbf{q}} = \frac{1}{n} \sum_{i=1}^{n} q_i \tag{1}$$

$$\mathrm{logNSE}(\mathbf{q}, \hat{\mathbf{q}}) = 1 - \frac{\sum_{i=1}^{n} (\log[q_i] - \log[\hat{q}_i])^2}{\sum_{i=1}^{n} (\log[q_i] - \log[\bar{q}])^2} \tag{2}$$

$$\mathrm{KGE}(\mathbf{q}, \hat{\mathbf{q}}) = 1 - \sqrt{(r-1)^2 + \left(\frac{\sigma_{\hat{\mathbf{q}}}}{\sigma_{\mathbf{q}}} - 1\right)^2 + \left(\frac{\mu_{\hat{\mathbf{q}}}}{\mu_{\mathbf{q}}} - 1\right)^2} \tag{3}$$

Where $\hat{q}_i \in \hat{\mathbf{q}}$ is the SWAT model prediction and $q_i \in \mathbf{q}$ is the observational streamflow. n is the dimension of $\hat{\mathbf{q}}$ and $\mathbf{q}$, which is the total number of time-steps. r is the Pearson product-moment correlation coefficient. $\sigma_{\hat{\mathbf{q}}}$ and $\sigma_{\mathbf{q}}$ are the standard deviations in the SWAT model predictions and observations, respectively. $\mu_{\hat{\mathbf{q}}}$ and $\mu_{\mathbf{q}}$ are the mean values in the SWAT model predictions and observations, respectively. The objective functions for computing mKGE and npKGE metrics are described in References (Kling et al., 2012; Pool et al., 2018).

Each metric takes into account different aspects of calibration performance (Liu, 2020). The R^2-score indicates the goodness of fit, which measures how close the streamflow predictions from the CNN-enabled calibration are to observed data. NSE evaluates how well the calibrated SWAT model predictions capture high flows. Complementary to NSE, logNSE determines the accuracy of model predictions for low flows. KGE combines these three different components of NSE (i.e., 1) correlation, 2) bias, and 3) a ratio of variances or coefficients of variation) in a more balanced way (e.g., more weight on low flows and less weight on extreme flows) to assess the SWAT model calibration. mKGE makes sure the bias and variability ratios are not cross-correlated, which otherwise may occur when (for instance the precipitation) inputs are biased. npKGE provides the variability and the correlation term in KGE in a non-parametric form. This reformulation of

KGE as npKGE allows us to estimate non-parametric components (i.e., the Spearman rank correlation and the normalized flow–duration curve), which are necessary for watershed model calibrations aiming at multiple hydrograph aspects. Hence, including multiple accuracy metrics when evaluating a calibrated model has obvious advantages. In addition to the above metrics, we quantify the uncertainty of the modeled streamflow for each method. Uncertainty is measured by the averaged width of maximum and minimum modeled streamflow results over the simulation periods and how well the modeled uncertainty boundary contains the observed streamflow. We evaluate this predictive uncertainty and associated probability that streamflow is contained within this boundary for the top-50 sets estimated by the CNN, DDS, and GLUE methods.

4.5 Computational cost

The wall clock time to run the WY 2014 to 2016 SWAT model simulation (each realization) is approximately an hour on a four-core processor (Intel(R) i7-8650U CPU at 1.90 GHz), which is a standard desktop computer. The ensemble run simulations for training the CNN-enabled inverse models were developed using a cluster of 56 cores (Intel(R) Xeon(R) Gold 5120 CPU at 2.20 GHz) and 256 GB DDR4 RAM. We trained a total of 26,250 CNN architectures by using 400 Cori's KNL CPU nodes at NERSC. Each KNL node comprises of 68 physical 1.40 GHz Intel Xeon Phi Processor 7,250 (Knights Landing) with four threads per core, 96 GB DDR4, and 16 GB MCDRAM memory. The scalable hyperparameter tuning to identify top-50 CNN architectures led to the use of approximately 520,000 processor hours. This is the total computational cost to calibrate the SWAT process model using CNNs. The time to calibrate SWAT process model using DDS and GLUE is equal to 20,000 processor hours (5,000 realizations × 4 cores). Even though model calibration using CNN is expensive (≈ 18 hours/architecture), it is embarrassingly parallel, allowing us to efficiently use supercomputing resources. The DDS method is generally sequential, as the parameter update depends on the previous estimation. Also, the DDS-based estimations depend on initial random guesses similar to CNN training. Hence, the algorithm needs to be run multiple times to remove the effect of randomness. Like the GLUE method, DDS allowed us to calibrate parameters in approximately 10 h.

From this training time, it is evident that a thorough hyperparameter tuning can be computationally expensive and requires high-performance computing resources. This high training time is mainly due to the slow training of CNN models on CPUs, which can be accelerated by using graphic processing units (GPUs). Despite the expensive computational cost to develop the proposed CNN-enabled inverse models, the inference cost to estimate the SWAT model parameters takes only 0.16 s. Moreover, hyperparameter tuning allows us to find CNNs that are highly accurate. The tuned CNNs allow us to make ensemble estimations quickly without the need to retrain the model. The GLUE and DDS algorithms need to be re-run on each discharge input to estimate SWAT parameters, which makes the trained CNNs attractive for inference. This low inference time is attractive for estimating the SWAT model parameters using streamflow data (with and without observational errors/noise). The wall clock time for making a prediction/inference shows that our CNN-enabled parameter estimation is at least $\mathcal{O}(10^3)$ times faster than the GLUE and DDS-based methods (e.g., may require thousands of forward model runs for each observational time series), in addition to its predictive capability.

5 Results

This section presents results on the overall accuracy and efficiency of the proposed DL methodology. First, we provide results from sensitivity analysis performed using a mutual information theory on the SWAT ensembles (Jiang et al., 2022b). Second, we describe the CNN-enabled inverse modeling results from the ensemble runs. Third, we show the SWAT model parameters estimated from observed discharges and compare the performance of CNN-enabled parameter estimation with the results from application of the GLUE and DDS methods. We also compare the streamflow predictions from the calibrated SWAT model with observed discharges for both the calibration and validation periods for all three methods. Finally, we give the performance metrics and calibration uncertainties for CNN-enabled, DDS, and GLUE estimated parameters.

5.1 Sensitivity analysis results

Table 1 identifies sensitive parameters that influence simulated discharge at the ARW study site. Figure 5 shows that the simulated discharge is sensitive to 11 out of the 20 parameters: 1) SFTMP, 2) CH_K2, 3) ALPHA_BF, 4) RCHRG_DP, 5) CH_N2, 6) SMTMP, 7) TIMP, 8) CN, 9) SMFMX, 10) GW_DELAY, and 11) EPCO. These 11 parameters correspond to landscape, groundwater, channel, plant, and snow groups. The important parameters mentioned above are identified using the MI methodology as described in (Cover and Thomas, 2006; Jiang et al., 2022b). Mutual information is a non-negative value that measures the dependency between the SWAT model parameters and its outputs. Zero MI means that streamflow is not affected by that parameter, and higher values of MI mean higher dependency. We note that discharge is primarily influenced

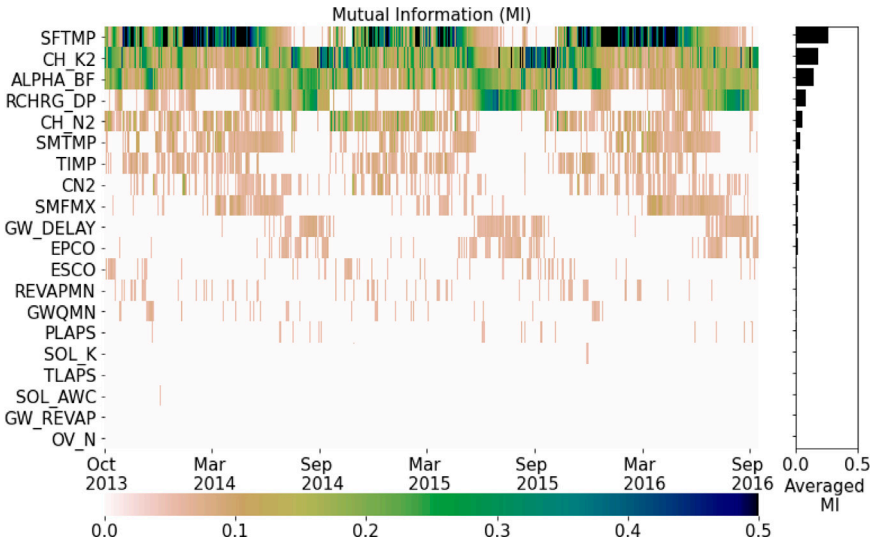

FIGURE 5
MI analysis on the SWAT model simulations: This figure shows the ranking of the SWAT model parameters based on MI. The analysis is performed for 1,000 realizations generated using a Sobol sequence. Among the sensitive parameters, it is evident that SFTMP is the dominant parameter and OV_N is the least important parameter for this ARW study site. In addition to snow parameters, we also see channel and groundwater parameter types are sensitive to streamflow.

by the snowfall temperature (SFTMP; the most sensitive), whose sensitivity shows the seasonality pattern consistent with the site description in Section 3.1. The importance of SFTMP in determining streamflow verifies the critical role of the snow process in this watershed.

5.2 Training, validation, and testing results

Figure 6A shows the training and validation loss of the best CNN-enabled inverse model in estimating the SWAT parameters. Validation loss plateaus even as the training loss decreases due to the lack of valuable information in the streamflow data to constrain the lesser and insensitive parameters (e.g., soil, climate, and other groundwater variables such as GW_REVAP). Figures 6B–D shows the prediction of the tuned CNN-enabled inverse model for estimating SFTMP. Supplementary Figures S1–S3 provide one-to-one plots for the remaining 10 sensitive parameters. Some one-to-one plots between the estimated and true parameters are closely distributed along the one-to-one line, which shows that the most sensitive parameters (e.g., SFTMP, CH_K2, ALPHA_BF) are predicted with reasonably good accuracy. The accuracy of the training predictions is lower for other less sensitive parameters (e.g., RCHRG_DP, EPCO). This reduced accuracy is evident from the more scattered drift away from the one-to-one straight line, seen in Supplementary Figure S3. The reduced accuracy is comparable to the training results where we see an increased deviation of data scatters from the one-to-one straight line. Similar results are obtained for other tuned CNN architectures. This scattered deviation indicates that these less-sensitive parameters are hard to predict using the discharge time series.

5.3 Sensitivity of estimated SWAT parameters to observation noise

We selected all test realizations to evaluate the parameter estimation sensitivity of the CNN-enabled inverse models to observed errors. We added random observation errors to the synthetic observed discharge time series for each test realization. We then generated 100 different observation realizations for parameter estimation, $\mathbf{q}_n$, which is given by (Cromwell et al., 2021)

$$\mathbf{q}_n = \mathbf{q} + \epsilon \times \mathbf{q} \times \mathbf{r} \tag{4}$$

where ϵ is the standard deviation of the noise, usually taken as $\frac{1}{3}$ of the observation error, and $\mathbf{r}$ is a random vector of the same size as $\mathbf{q}$. The elements of the random vector contain samples drawn from a standard normal distribution with a mean of 0 and a standard deviation of 1. We tested different levels of observation errors (i.e., 5%, 10%, 25%, 50%, and 100%) relative to the observed values. These noisy discharge data (both synthetic and observations) are provided as input to the best CNN-enabled inverse models to estimate the SWAT model parameters.

Figure 7 shows the variability in estimated SFTMP results from the CNNs results as box plots. It also shows CNN model predictions for all noisy test realizations (Figure 7A) as well as

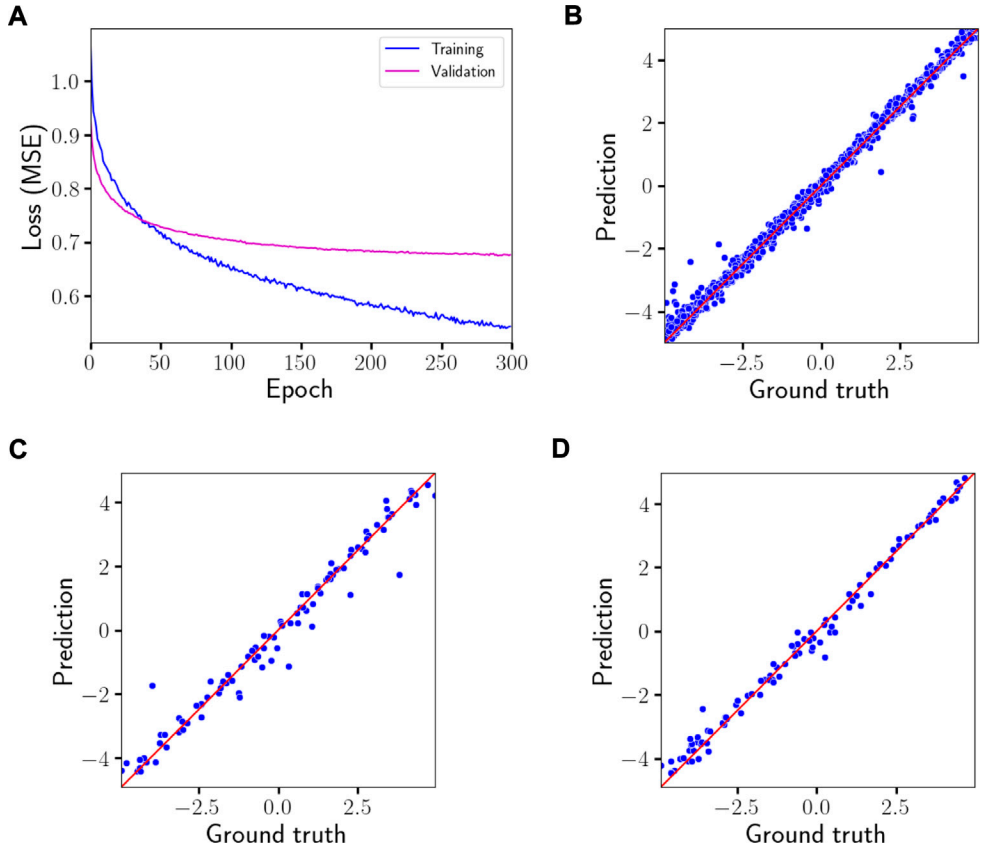

FIGURE 6
Loss metrics of the best CNN-enabled inverse model and its predictions for the SFTMP: The top left figure **(A)** figure shows the overall training and validation loss of the best CNN architecture. The top right **(B)**, lower left **(C)**, and lower right **(D)** figures show one-to-one plots for the most sensitive parameter, SFTMP (units in °C). It compares the CNN estimation with the ground truth for the training, validation, and test datasets. We did not use test data for finding the tuned CNN architectures. Only the validation set is used for hyperparameter tuning. Each blue dot represents a realization from the corresponding train/validation/test set of ensembles. The red line is the one-to-one line.

observed data (Figure 7B). Supplementary Figures S4–S8 provide estimates for the other sensitive parameters. We note that the parameter estimates are within the prior sampling range even after adding high relative noise levels (i.e., 100%), which instills confidence in the predictive capabilities of CNN models. From Figure 7A, it is evident that CNN-enabled inverse models are reasonably robust to noise in estimating the most sensitive parameter. This shows that the SFTMP predictions are not that sensitive to noise, as the performance of CNNs is stable even after adding high errors to data. This predictive capability when noise is applied to discharge time series also provides the insight that CNNs can effectively learn underlying representations in the streamflow data rather than noise in the observed data. Similar assessments can be made for the other sensitive parameters (e.g., CH_K2, ALPHA_BF). However, as model parameter sensitivity decreases, the CNN predictions are more prone to be influenced by noise. The performance of CNN estimation for EPCO, which is the least sensitive parameter among the top 11 parameters, is lower than that of sensitive parameters such as SFTMP. As discussed in Section 5.2 and from MI analysis, it is evident that streamflow provides little information to estimate this parameter. This reduced performance is the result of less valuable information being available in the streamflow data to accurately estimate less sensitive parameters, such as EPCO.

5.4 Calibrated SWAT model based on observed discharge

Trained CNN-enabled inverse models are used to estimate the SWAT parameters at the ARW study site based on observed discharge data. We provide streamflow predictions of the calibrated SWAT model based on the best CNN architecture and the following 49 best candidates. We also compare the

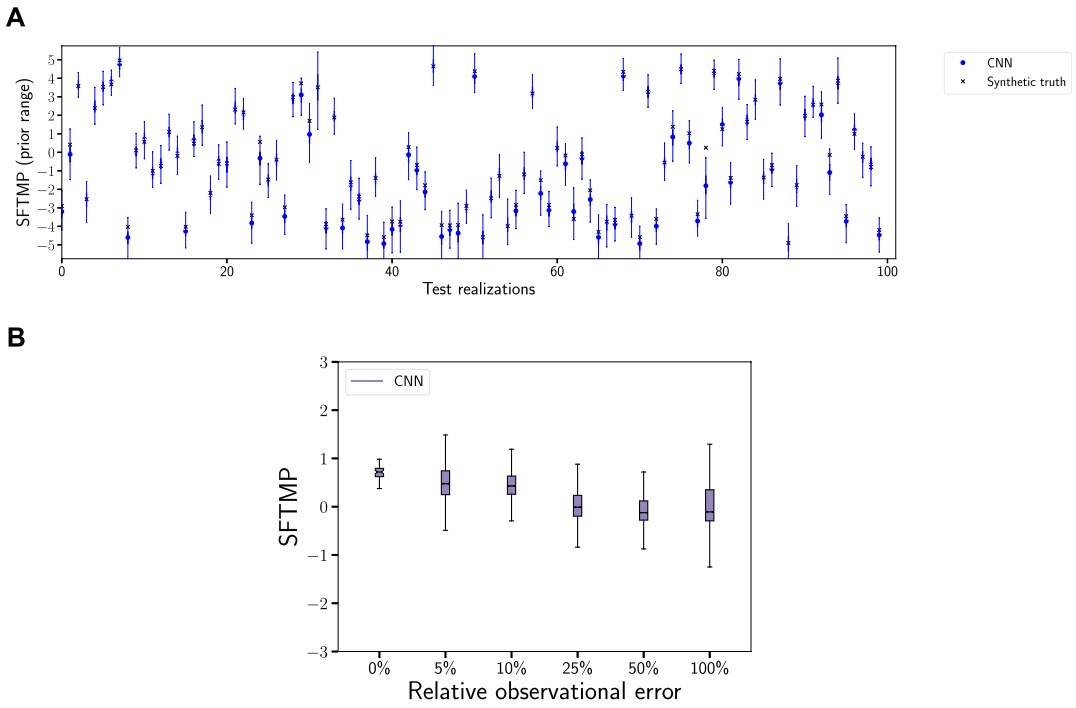

FIGURE 7
Top-50 CNN model estimation under influence of noise (SFTMP): The top figure **(A)** shows the sensitivities of the CNN-enabled inverse models to 100% noise added to the test realizations. The bottom figure **(B)** shows the CNN estimations on the observational discharge represented by the colors filling the box plots. Both of these figures show the variation in SFTMP (units in °C) with different noise levels. Moreover, the CNN estimations are closer to ground truth for all synthetic predictions as shown in the top figure. From the bottom figure, we observe that even under high relative observational errors, the estimations of the most sensitive parameter SFTMP are narrower compared to GLUE as seen in Figure 8.

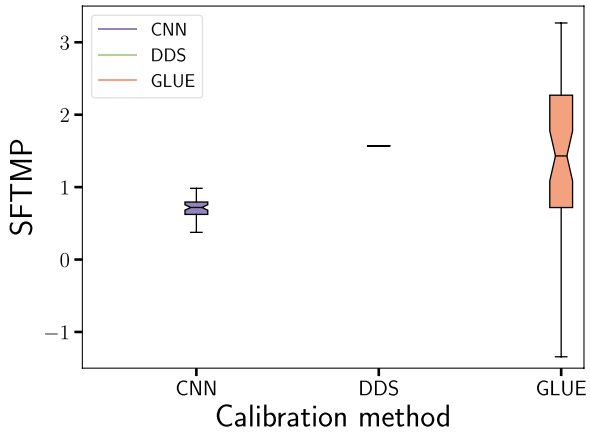

FIGURE 8
Comparison of top-50 CNN, DDS, and GLUE estimated sets (SFTMP): This figure compares the estimations of CNN, DDS, and GLUE on observational discharge for SFTMP (units in °C). From this figure, it is evident that DDS has the narrower range than CNN and GLUE.

performance of CNN predictions against predictions provided by the DDS and GLUE methods. Figure 8 shows the estimated SFTMP parameter range for CNN, DDS, and GLUE for the observed data. We see that the DDS method has a narrower range than the CNN and GLUE methods. The reason for this narrower range is DDS uses a global optimization algorithm that iteratively searches for a parameter set that produces a unique value. On the other hand, the loss function in the CNN method is non-convex, meaning that in all likelihood, gradient descent converges to sub-optimal valleys or local minima. Hence, the CNN method has a slightly broader range compared to the DDS method but a narrower range than the GLUE method. Similar inferences can be made on other parameters as shown in Supplementary Figure S9 provided in the supplementary information (e.g., ALPHA_BF, CH_K2, RCHRG_DP).

Figure 9 shows the calibration performance of top-50 CNN, DDS, and GLUE calibration set predictions using six different metrics. It is clear that CNN-enabled parameter estimation is better than behavioral parameter sets estimated by the GLUE and DDS methods for all six studied metrics. Additionally, in

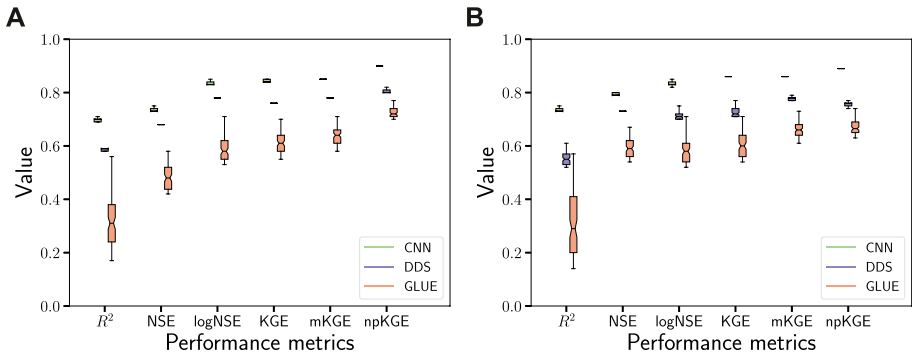

FIGURE 9
Performance metrics of top-50 estimated sets using CNN, DDS, and GLUE methods: This figure compares different performance metrics of the CNN, DDS, and GLUE calibrated sets. The left **(A)** and right **(B)** figures show the performances in calibration and validation periods, respectively. The green, blue, and red whiskers represent the CNN estimation, DDS, and GLUE. Top-50 best performance sets are identified and evaluated for each method within and beyond calibration period. The performance metrics (e.g., NSE, logNSE, npKGE) focus on the predictive capability of CNN-, DDS-, and GLUE-based calibrated SWAT models in both low and high flow scenarios. Across all performance metrics, it is evident that estimation using the CNN-enabled inverse models outperforms DDS and GLUE.

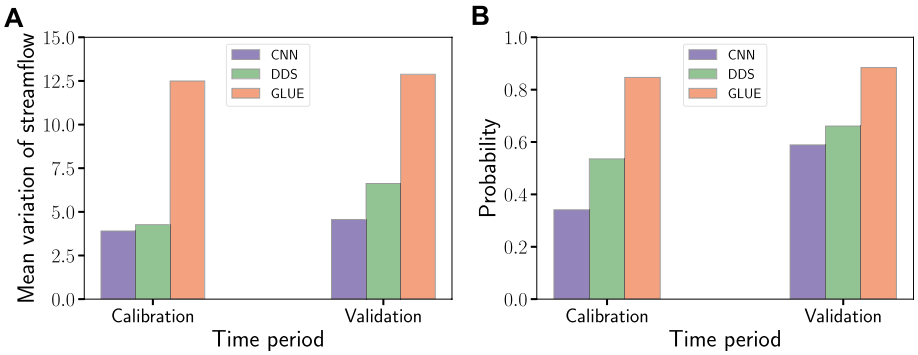

FIGURE 10
Comparison of top-50 CNN, DDS, and GLUE's streamflow variations: The left figure **(A)** shows the size of the mean modeled streamflow variation (i.e., a representation of predictive uncertainty). The right figure **(B)** provides the probability that the observed flow is contained within the predicted bounds of the streamflow (e.g., the light blue colored region in Figure 11) estimated by the calibrated SWAT model. The uncertainty in the GLUE-based calibration sets prediction, and associated probability is higher than DDS and CNN.

Supplementary Figure S15, we show one-to-one scatter plots for the best CNN, DDS, and GLUE streamflow predictions with observed data both in calibration and validation periods. In Supplementary Figure S15, each dot corresponds to daily streamflow. The predictions are based on the best sets calibrated by the CNN, GLUE, or DDS methods. The best CNN-based calibrated set has R^2, NSE, logNSE, KGE, mKGE, and npKGE scores of 0.71, 0.75, 0.85, 0.85, 0.86, and 0.91, respectively. The best DDS-based calibrated set has scores of 0.62, 0.69, 0.8, 0.77, 0.79, and 0.82. The best GLUE-based calibrated set has scores of 0.56, 0.58, 0.71, 0.7, 0.71, and 0.8. Supplementary Table S2 also provides the metric values for the CNN, GLUE, and DDS sets. From these values, it is clear that the CNN-enabled inverse model estimations are more accurate for SWAT model calibration than the GLUE and DDS estimations. Therefore, CNNs show promise for parameter estimation, especially in nonlinearly relating streamflow data to conceptual parameters.

Figure 10 shows smaller uncertainty ranges for CNN sets in both calibration and validation periods than the GLUE and DDS estimations. The probability that the prediction intervals estimated by the CNN sets contain the observed streamflow also is lower than the GLUE and DDS sets. This shows that top-50 CNN sets are not sufficient to capture the predictive boundary of streamflow variations. If we include all the CNN sets (as shown in Supplementary Figures S12–S15G,H), the probability that

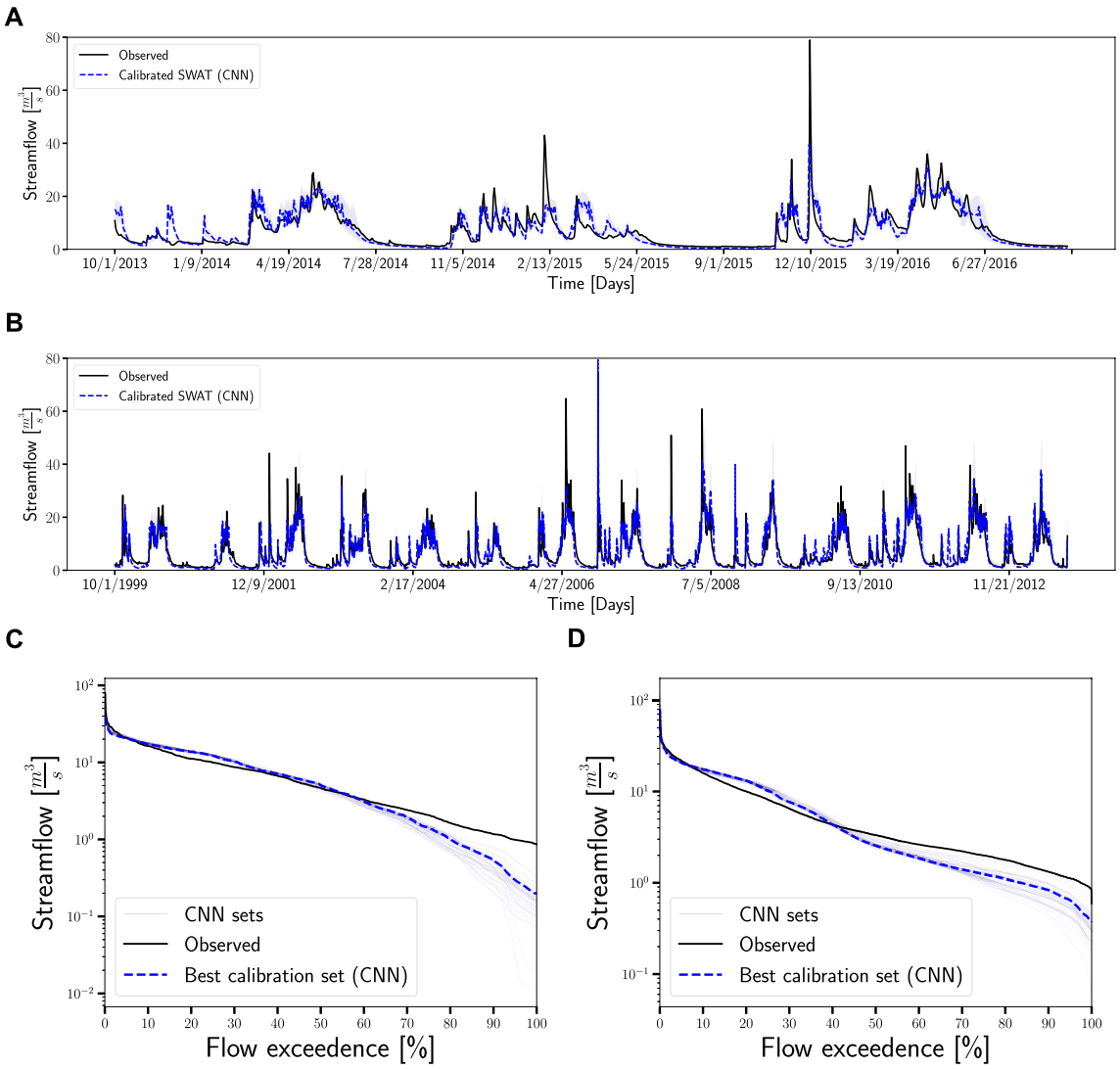

FIGURE 11
Comparison of the calibrated SWAT model (top-50 CNN) with observation data: This figure **(A)** and **(B)** compares the predictions of the calibrated SWAT model with observational data within and beyond the calibration period. The solid black line represents the observational data. The dashed colored (blue) line represents the predictions based on the best calibrated set using CNN. The light colored region in the streamflow plots represents the prediction uncertainty. This region is calculated by running the SWAT model using the calibration sets obtained using CNN. The bottom figures **(C)** and **(D)** show the flow duration curves in both calibration and validation period. It is clear that CNN estimation sets produce curves that are reasonably closer to observational data.

observational data is contained within the prediction bounds is greater than 0.95. However, the mean variation size of the streamflow increases fourfold to accommodate this increase in probability. One of our next steps is to improve this top-50 CNN estimation probability while keeping predictive uncertainty low. This can be achieved through ensemble DL, knowledge-guided DL, and probabilistic BNNs (Lu et al., 2021; Jiang et al., 2022a). These types of networks can account for uncertainty so that CNN-enabled inverse models can assign lower confidence levels to incorrect predictions. Figure 11 compares the streamflow predictions from the calibrated SWAT with the observed data using the top-50 CNN model sets. Supplementary Figures S10, S11 shows the predictions from both the top-50 GLUE and DDS methods. The CNN estimations capture the various high and low streamflows better than the GLUE and DDS methods during both the calibration and validation periods. However, the calibrated SWAT model over predicts in certain parts of WY 2014 (e.g., 9 January 2014) and WY 2015. This lower predictive performance may imply potential deficiencies (i.e., structural errors) in the underlying SWAT model representation of

watershed processes. Additional investigations are necessary to identify other processes and parameters that reduce structural errors and discrepancies in streamflow predictions.

6 Possible extensions of current work

Our results demonstrate the applicability of using scalable deep learning to calibrate the SWAT model. We note that the proposed methodology is general and can be used to calibrate other watershed models such as ATS and PRMS. This extensibility for calibrating other models and study sites can be achieved using transfer learning methods (Zhuang et al., 2020), which will allow us to reuse the CNNs developed in this study and leverage them for new, similar problems. Minimal re-training is necessary to fine tune the trained CNNs (Song and Tartakovsky, 2021) and apply them to calibrate watershed models for other study sites. Such a transfer of knowledge across study sites usually is performed when generating the large amount of training data needed to develop a full-scale CNN and tuning its trainable weights from the start is too computationally expensive (e.g., when using ATS). Additionally, we can improve our DL methodology to calibrate the SWAT model by incorporating other multi-source data streams (e.g., evapotranspiration (ET), and snow water equivalent (SWE)) along with streamflow. Our next step is to use such data streams to further investigate the deficiency of the model structure or processes in the SWAT model by ingesting streamflow, ET, and SWE into CNNs.

Figure 10B shows the DL method's probability that observational data contained within the prediction bounds are lower than probabilities provided by the DDS and GLUE methods. There are multiple ways in which we can improve our CNN-based parameter estimation and predictive uncertainty. A possible approach involves accelerating the training process using GPUs available at leadership class supercomputing resources (e.g., NERSC, Oak Ridge Leadership Computing Facility, and Argonne Leadership Computing Facility user facilities) (ALCF, 2021; NERSC, 2021; OLCF, 2021). This accelerated CNN training allows us to developed ensemble learning models through bootstrapping, which are known to provide better generalization performance than a final CNN.

As discussed in Section 5.4, improved uncertainty intervals can be achieved through ensemble learning (e.g., combining predictions of different types of neural networks such as DNNs and BNNs). Additionally, developing CNNs tailored to estimate the SWAT model parameters under different hydrological seasons (McMillan, 2020) (e.g., winter vs. summer) may enhance the calibration process. For example, comparing CNN-estimated sets from wet and dry periods of the year can provide better insights into the SWAT model parameters that control streamflow predictions across different seasons. When making such comparisons between real data and model predictions, hydrological signatures and their associated metrics (Westerberg and McMillan, 2015; McMillan et al., 2017; Fatehifar et al., 2021; Gnann et al., 2021; McMillan, 2021) can be used to elucidate the structural deficiencies of the SWAT model. Hydrological signatures on which we can evaluate performance metrics include the slope of the flow duration curve, rising limb density, recession shape, and baseflow index of streamflow time-series data (McMillan, 2021).

In addition to the data-driven methodology presented in this paper[6], the efficacy of the proposed DL methodology also can be improved by embedding domain knowledge into DNNs (Read et al., 2019; Khandelwal et al., 2020; Bhasme et al., 2021; Jia et al., 2021). Recent advances in knowledge-guided machine learning (Jiang et al., 2022a) provide a way to incorporate model states/fluxes and water balances as part of recurrent neural network architectures (Khandelwal et al., 2020). The papers mentioned above used such neural architectures to develop forward emulators for watershed models. One can extend the methods presented in those works to incorporate process model knowledge into our proposed CNNs to improve SWAT model calibration. Also, Explainable AI (XAI) methods such as deep Taylor decomposition (Kindermans et al., 2016), SHAPley values (Messalas et al., 2019), and integrated gradients (Sundararajan et al., 2017) can be used to explain the CNN predictions. These XAI methods not only allow us to explain why CNNs provide results that are understandable for the domain experts (Leduc et al., 2020) but also extract informative signals (e.g., precursors) from the streamflow time-series data (McMillan et al., 2017; McMillan, 2020; McMillan, 2021).

7 Conclusion

In this paper, we describe an accurate and reliable DL methodology that we developed to calibrate the SWAT model. We used CNN-enabled inverse models to estimate the SWAT parameters for the ARW study site in the YRB. Our approach leverages recent advances in CNNs to extract representations from streamflow data and then map them to the SWAT model parameters. Scalable hyperparameter tuning was performed to identify optimal CNN architectures. Ensemble runs from the SWAT model were used to train, validate, and test the CNN-enabled inverse models. We performed sensitivity analyses to identify the dominant parameters that influence streamflow. Our results show that CNN models are able to estimate the sensitive

6 Or by combining Markov chain Monte Carlo methods with forward emulators (Dagon et al., 2020) for model calibration.

parameters reasonably well. The parameters estimated from the trained CNNs were robust to high observed errors. We then compared the SWAT parameters estimated by our DL method with parameters generated by the GLUE and DDS optimization algorithms. We found that all the methods estimated SWAT parameters within the sampling range of the ensemble runs. As DDS is a global optimization method, its estimated range of parameters are narrower compared to parameters estimated by the GLUE and DL methods. Furthermore, this comparison also showed that predictions of the calibrated SWAT model based on CNNs performs better than the GLUE and DDS methods. Key performance metrics (e.g., R^2-score, NSE, logNSE, KGE, and its variants) showed that the best CNN-based calibration sets capture low and high flows better than the GLUE and DDS methods. This improvement in predictive performance is probably because CNNs can more effectively use the information (e.g., learning representative features from streamflow) provided in ensemble runs than the GLUE and DDS methods. By capturing the nonlinear relationships between SWAT model inputs and outputs through multiple convolutional neural layers, CNNs yielded more realistic predictions for the ARW and a better calibrated SWAT model. This improvement resulted in a closer match between model-predicted and observed stream discharges. Our results showed that the probability that the observed data are contained within the prediction bounds estimated by top-50 CNN sets is lower than that of DDS and GLUE sets. This lower probability shows that the top-50 CNN sets alone are insufficient to capture the variations in streamflow. If all the CNN estimations are included, we are able to capture the observed data within the prediction bounds. However, including all the CNN estimations resulted in higher mean variation of streamflow (i.e., fourfold increase when compared to the top-50 CNN sets). Our future work involves further improving the accuracy the CNN method while keeping the predictive uncertainty (i.e., size of streamflow variation) lower.

From a computational cost perspective, the time needed to infer parameters based on the DL method is at least $\mathcal{O}(10^3)$ faster than that of the GLUE and DDS methods, which makes extending this method to complex watershed models (e.g., ATS) attractive. However, the computational cost of identifying optimal CNN architectures is high compared to the GLUE and DDS methods. The training time needed to develop CNN models can be improved further by using GPUs and TPUs (Bisong, 2019). Reducing the computational cost of developing CNN-enabled inverse models is one of our next steps, with a focus on using the distributed deep learning training framework (e.g., using Horovod (Sergeev and Del Balso, 2018) or DeepHyper (Balaprakash et al., 2018)) that already shows promise in the training speedup. This improves the efficiency during training process by using asynchronous distributed Bayesian optimization algorithms, which are known to be much more efficient than the grid search that has to exhaust all the hyperparameter space.

Our methodology is general and can be used to calibrate complex watershed models (i.e., through transfer learning methods (Zhuang et al., 2020)) with minimal re-training. For example, using transfer learning. Transfer learning consists of using pre-trained deep learning models such as CNNs on one watershed and leveraging them on a new and similar watershed. Specifically, transfer learning (Oruche et al., 2021) allows us to transfer knowledge from gauged (e.g., ARW) to ungauged basins (e.g., YRB) or watersheds (Westerberg et al., 2016; Guo et al., 2021). This knowledge transfer is usually done when training a full-scale CNN from scratch is challenging due to the availability of limited simulation data or when regions are data sparse, observationally. In such scenarios, a watershed classification scheme is first used to identify a new watershed with characteristics similar to ARW. Then, the neural features from the pre-trained CNN that has learned to extract patterns from ARW's streamflow data can be adapted to that new paired watershed. Finally, fine-tuning is performed to achieve meaningful improvements by incrementally adapting the pre-trained CNN's features to the new simulation data. For fine-tuning to be successful, minimal simulation data on the newly selected watershed is needed. Additional future work involves modifying the proposed method to incorporate multi-source datasets (e.g., by combining streamflow, ET, and SWE) to further enhance SWAT model calibration (Moriasi et al., 2007; Samimi et al., 2020), and transfer the knowledge gained on ARW to the entire Yakima river basin (i.e., by transfer learning).

Data availability statement

The data generated for the proposed DL model development uses open-science principles. Specifically, we make the data Findable Accessible Interoperable and Reusable (FAIR). FAIR principles expedite community-based data generation, modeling, and interdisciplinary collaboration and provides a means to test new hypotheses. The datasets generated and analyzed as well as scripts for this study, will be made available on this GitHub repository: ˜\url {https://github.com/maruti-iitm/DL4SWAT.git} upon publication. \texttt{SWAT} open-source code can be downloaded at \url {https://swat.tamu.edu/}.

Author contributions

MM: Conceptualization, methodology, software, data curation, visualization, investigation, writing—original draft, writing—review and editing. KS: Methodology, data generation, writing—review and editing. PJ: Sensitivity analysis, writing—review and editing. GH: Methodology,

writing—review and editing. XC: writing—review and editing and funding.

Funding

This research was supported by the U.S. Department of Energy (DOE), Office of Science (SC) Biological and Environmental Research (BER) program, as part of BER's Environmental System Science program.

Acknowledgments

This contribution originates from the River Corridor Scientific Focus Area at Pacific Northwest National Laboratory (PNNL). This research used resources from the National Energy Research Scientific Computing Center, a DOE-SC User Facility. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. The authors thank the reviewers whose feedback helped in substantially improving the manuscript.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/feart.2022.1026479/full#supplementary-material

References

Abbaspour, K. C. (2013). *Swat-cup 2012. SWAT calibration and uncertainty program–A user manual.*

Adams, B. M., Bohnhoff, W. J., Dalbey, K. R., Eddy, J. P., Eldred, M. S., Gay, D. M., et al. (2009). *Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 user's manual*. Tech. Rep. SAND2010-2183.*Sandia Natl. Lab.*

Afzaal, H., Farooque, A. A., Abbas, F., Acharya, B., and Esau, T. (2020). Groundwater estimation from major physical hydrology components using artificial neural networks and deep learning. *Water* 12, 5. doi:10.3390/w12010005

ALCF (2021). *Argonne leadership computing facility*. Available at: https://www.alcf.anl.gov/ (Accessed on 07, 202121).

Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., et al. (2009). The data assimilation research testbed: A community facility. *Bull. Am. Meteorol. Soc.* 90, 1283–1296. doi:10.1175/2009bams2618.1

Anysz, H., Zbiciak, A., and Ibadov, N. (2016). The influence of input data standardization method on prediction accuracy of artificial neural networks. *Procedia Eng.* 153, 66–70. doi:10.1016/j.proeng.2016.08.081

Arnold, J. G., Moriasi, D. N., Gassman, P. W., Abbaspour, K. C., White, M. J., Srinivasan, R., et al. (2012). Swat: Model use, calibration, and validation. *Trans. ASABE* 55, 1491–1508. doi:10.13031/2013.42256

Aster, R. C., Borchers, B., and Thurber, C. H. (2018). *Parameter estimation and inverse problems*. Elsevier.

Bacu, V., Nandra, C., Stefanut, T., and Gorgan, D. (2017). SWAT model calibration over Cloud infrastructures using the BigEarth platform. *13th IEEE Int. Conf. Intelligent Comput. Commun. Process. (ICCP)*, 453–460.

Balaprakash, P., Salim, M., Uram, T., Vishwanath, V., and Wild, S. (2018). Deephyper: Asynchronous hyperparameter search for deep neural networks. *IEEE 25th Int. Conf. high Perform. Comput. (HiPC)*, 42–51.

Beven, K., and Binley, A. (2014). Glue: 20 years on. *Hydrol. Process.* 28, 5897–5918. doi:10.1002/hyp.10082

Bhasme, P., Vagadiya, J., and Bhatia, U. (2021). *Enhancing predictive skills in physically-consistent way: Physics informed machine learning for hydrological processes*. arXiv preprint arXiv:2104.11009.

Bisong, E. (2019). "Google colaboratory," in *Building machine learning and deep learning models on google cloud platform* (Berlin: Springer), 59–64.

Blasone, R.-S., Vrugt, J. A., Madsen, H., Rosbjerg, D., Robinson, B. A., and Zyvoloski, G. A. (2008). Generalized likelihood uncertainty estimation (GLUE) using adaptive Markov chain Monte Carlo sampling. *Adv. Water Resour.* 31, 630–648. doi:10.1016/j.advwatres.2007.12.003

Chen, X., Hammond, G. E., Murray, C. J., Rockhold, M. L., Vermeul, V. R., and Zachara, J. M. (2013). Application of ensemble-based data assimilation techniques for aquifer characterization using tracer data at Hanford 300 area. *Water Resour. Res.* 49, 7064–7076. doi:10.1002/2012wr013285

Chen, X., Shuai, P., Son, K., Jiang, P., Mudunuru, M., Coon, E., et al. (2021). AGU fall meeting abstracts. In *What can we learn from multiple watershed models and observations?* 2021. H23H–01.

Chiang, L.-C., and Yuan, Y. (2015). The NHDPlus dataset, watershed subdivision and SWAT model performance. *Hydrological Sci. J.* 60, 1690–1708. doi:10.1080/02626667.2014.916408

Chollet, F. (2017). *Deep learning with Python*. Shelter Island, NY: Manning Publications Company.

Coon, E. T., Berndt, M., Jan, A., Svyatsky, D., Atchley, A. L., Kikinzon, E., et al. (2020). *Advanced terrestrial simulator*. USA: U.S. Department of Energy. Version 1.0. doi:10.11578/dc.20190911.1

Cover, T. M., and Thomas, J. A. (2006). *Wiley Series in Telecommunications and Signal Processing*. Elements of information theory.

Cromwell, E. L. D., Shuai, P., Jiang, P., Coon, E., Painter, S. L., Moulton, D., et al. (2021). Estimating watershed subsurface permeability from stream discharge data using deep neural networks. *Front. Earth Sci. (Lausanne).* 9. doi:10.3389/feart.2021.613011

Cuo, L., Lettenmaier, D. P., Mattheussen, B. V., Storck, P., and Wiley, M. (2008). Hydrologic prediction for urban watersheds with the distributed hydrology–soil–vegetation model. *Hydrol. Process.* 22, 4205–4213. doi:10.1002/hyp.7023

Dagon, K., Sanderson, B. M., Fisher, R. A., and Lawrence, D. M. (2020). A machine learning approach to emulation and biophysical parameter estimation with the Community Land Model, version 5. *Adv. Stat. Climatol. Meteorol. Oceanogr.* 6, 223–244. doi:10.5194/ascmo-6-223-2020

Daly, C., and Bryant, K. (2013). *The PRISM climate and weather system–An introduction*. Corvallis, OR: PRISM climate group.

Daly, C., Taylor, G. H., Gibson, W. P., Parzybok, T. W., Johnson, G. L., and Pasteris, P. A. (2000). High-quality spatial climate data sets for the United States and beyond. *Trans. ASAE* 43, 1957–1962. doi:10.13031/2013.3101

Daniel, E. B., Camp, J. V., LeBoeuf, E. J., Penrod, J. R., Dobbins, J. P., and Abkowitz, M. D. (2011). Watershed modeling and its applications: A state-of-the-art review. *Open Hydrology J.* 5, 26–50. doi:10.2174/1874378101105010026

Daymet (2021). *Daily surface weather and climatological summaries*. Available at: https://daymet.ornl.gov/ (Accessed on 07, 202121).

Doherty, J. E., and Hunt, R. J. (2010). *Approaches to highly parameterized inversion: A guide to using PEST for groundwater-model calibration*, 2010. Middleton, WI: U.S. Geological Survey.

Donigian, A. S., Jr, Bicknell, B. R., and Imhoff, J. C. (1995). Hydrological simulation program-fortran (HSPF). *Comput. models watershed hydrology*, 395–442.

Duan, Q., Gupta, H. V., Sorooshian, S., Rousseau, A. N., and Turcotte, R. (2004). *Calibration of watershed models*. American Geophysical Union.

Duan, Q., Sorooshian, S., and Gupta, V. K. (1994). Optimal use of the SCE-UA global optimization method for calibrating watershed models. *J. Hydrology* 158, 265–284. doi:10.1016/0022-1694(94)90057-4

Eckhardt, K., Fohrer, N., and Frede, H.-G. (2005). Automatic model calibration. *Hydrol. Process.* 19, 651–658. doi:10.1002/hyp.5613

Edwards, C. (2018). Deep learning hunts for signals among the noise. *Commun. ACM* 61, 13–14. doi:10.1145/3204445

Evensen, G. (2018). Analysis of iterative ensemble smoothers for solving inverse problems. *Comput. Geosci.* 22, 885–908. doi:10.1007/s10596-018-9731-y

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.* 99, 10143–10162. doi:10.1029/94jc00572

Evensen, G. (2003). The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean. Dyn.* 53, 343–367. doi:10.1007/s10236-003-0036-9

Fang, Y., Chen, X., Velez, J. G., Zhang, X., Duan, Z., Hammond, G. E., et al. (2020). A multirate mass transfer model to represent the interaction of multicomponent biogeochemical processes between surface water and hyporheic zones (SWAT-MRMT-R 1.0). *Geosci. Model. Dev.* 13, 3553–3569. doi:10.5194/gmd-13-3553-2020

Fatehifar, A., Goodarzi, M. R., Montazeri, H., S. S., and Dastjerdi, S. (2021). Assessing watershed hydrological response to climate change based on signature indices. *J. Water Clim. Change* 12, 2579–2593. doi:10.2166/wcc.2021.293

Franco, A. C. L., and Bonumá, N. B. (2017). Multi-variable SWAT model calibration with remotely sensed evapotranspiration and observed flow. *RBRH* 22. doi:10.1590/2318-0331.011716090

Gabrielli, L., Tomassetti, S., Squartini, S., and Zinato, C. (2017). "Introducing deep machine learning for parameter estimation in physical modelling," in *Proceedings of the 20th international conference on digital audio effects*.

Gnann, S. J., Coxon, G., Woods, R. A., Howden, N. J. K., and McMillan, H. K. (2021). Tossh: A toolbox for streamflow signatures in hydrology. *Environ. Model. Softw.* 138, 104983. doi:10.1016/j.envsoft.2021.104983

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

Graham, D. N., and Butts, M. B. (2005). Flexible, integrated watershed modelling with MIKE SHE. *Watershed models*. 849336090, 245–272.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognit.* 77, 354–377. doi:10.1016/j.patcog.2017.10.013

Guo, Y., Zhang, Y., Zhang, L., and Wang, Z. (2021). Regionalization of hydrological modeling for predicting streamflow in ungauged catchments: A comprehensive review. *WIREs Water* 8, e1487. doi:10.1002/wat2.1487

Gupta, H. V., Sorooshian, S., Hogue, T. S., and Boyle, D. P. (2003). Advances in automatic calibration of watershed models. *Calibration Watershed Models* 6, 9–28.

Gupta, S., and Gupta, A. (2019). Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Comput. Sci.* 161, 466–474. doi:10.1016/j.procs.2019.11.146

Hamman, J. J., Nijssen, B., Bohn, T. J., Gergel, D. R., and Mao, Y. (2018). The Variable Infiltration Capacity model version 5 (VIC-5): Infrastructure improvements for new applications and reproducibility. *Geosci. Model. Dev.* 11, 3481–3496. doi:10.5194/gmd-11-3481-2018

Herman, J., and Usher, W. (2017). SALib: An open-source Python library for sensitivity analysis. *J. Open Source Softw.* 2, 97. doi:10.21105/joss.00097

Hydroeval (2021). *An evaluator for streamflow time series in Python*. Available at: https://github.com/ThibHlln/hydroeval.git (Accessed on 0803, 2022).

Jagtap, N. V., Mudunuru, M. K., and Nakshatrala, K. B. (2021). A deep learning modeling framework to capture mixing patterns in reactive-transport systems. *Commun. Comput. Phys.* 0.4208/cicp.OA-2021-0088.

Jia, X., Willard, J. D., Karpatne, A., Read, J. S., Zwart, J. A., Steinbach, M., et al. (2021). Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM. IMS. Trans. Data Sci.* 2, 1–26. doi:10.1145/3447814

Jiang, P., Chen, X., Chen, K., Anderson, J., Collins, N., and Gharamti, M. E. (2021). DART-PFLOTRAN: An ensemble-based data assimilation system for estimating subsurface flow and transport model parameters. *Environ. Model. Softw.* 142, 105074. doi:10.1016/j.envsoft.2021.105074

Jiang, P., Shuai, P., Sun, A., Mudunuru, M. K., and Chen, X. (2022a). Knowledge-informed deep learning for hydrological model calibration: An application to coal creek watershed in Colorado. *Hydrology Earth Syst. Sci. Discuss.*, 1–31. doi:10.5194/hess-2022-282

Jiang, P., Son, K., Mudunuru, M. K., and Chen, X. (2022b). *Using mutual information for global sensitivity analysis on watershed modeling*. Malden, MA: John Wiley & Sons Inc.

Johnston, P. R., and Pilgrim, D. H. (1976). Parameter optimization for watershed models. *Water Resour. Res.* 12, 477–486. doi:10.1029/wr012i003p00477

Keras API (2021). *The high-level API of Tensorflow*. Available at: https://www.tensorflow.org/api_docs/python/tf/keras (Accessed on 07, 202121).

Khandelwal, A., Xu, S., Li, X., Jia, X., Stienbach, M., Duffy, C., et al. (2020). *Physics guided machine learning methods for hydrology*. arXiv preprint arXiv: 2012.02854.

Kindermans, P.-J., Schütt, K., Müller, K.-R., and Dähne, S. (2016). *Investigating the influence of noise and distractors on the interpretation of neural networks*. arXiv preprint arXiv:1611.07270.

Kling, H., Fuchs, M., and Paulin, M. (2012). Runoff conditions in the upper Danube basin under an ensemble of climate change scenarios. *J. hydrology* 424, 264–277. doi:10.1016/j.jhydrol.2012.01.011

Leavesley, G. H., Lichty, R. W., Troutman, B. M., and Saindon, L. G. (1983). Precipitation-runoff modeling system: User's manual. *Water-resources Investig. Rep.* 83, 207.

Leduc, R., Hulbert, C., McBrearty, I. W., and Johnson, P. A. (2020). Probing slow earthquakes with deep learning. *Geophys. Res. Lett.* 47, e2019GL085870. doi:10.1029/2019gl085870

Liu, D. (2020). A rational performance criterion for hydrological model. *J. Hydrology* 590, 125488. doi:10.1016/j.jhydrol.2020.125488

Lu, D., Konapala, G., Painter, S. L., Kao, S.-C., and Gangrade, S. (2021). Streamflow simulation in data-Scarce basins sing Bayesian and physics-informed machine learning models. *J. Hydrometeorol.* 22, 1421–1438.

MADS (2021). *Model analysis & decision Support*. Available at: https://mads.lanl.gov/ (Accessed on 07, 202121).

Mankin, D., Srinivasan, R., and Arnold, J. G. (2010). Soil and water assessment tool (SWAT) model: Current developments and applications. *Trans. ASABE* 53, 1423–1431. doi:10.13031/2013.34915

Marçais, J., and de Dreuzy, J.-R. (2017). Prospective interest of deep learning for hydrological inference. *Groundwater* 55, 688–692. doi:10.1111/gwat.12557

Markstrom, S. L., Regan, R. S., Hay, L. E., Viger, R. J., Webb, R. M. T., Payn, R. A., et al. (2015). PRMS-IV, the precipitation-runoff modeling system, version 4. *U. S. Geol. Surv. Tech. Methods* 6, B7.

Mastin, M. C., and Vaccaro, J. J. (2002). *Tech. Rep., open-file report 02-404*. Washington DC, USA: U.S. Department of Interior.Watershed models for decision support in the Yakima river basin, Washington

McMillan, H. K. (2021). A review of hydrologic signatures and their applications. *WIREs Water* 8, e1499. doi:10.1002/wat2.1499

McMillan, H. (2020). Linking hydrologic signatures to hydrologic processes: A review. *Hydrol. Process.* 34, 1393–1409. doi:10.1002/hyp.13632

McMillan, H., Westerberg, I., and Branger, F. (2017). Five guidelines for selecting hydrological signatures. *Hydrol. Process.* 31, 4757–4761. doi:10.1002/hyp.11300

Mein, R. G., and Brown, B. M. (1978). Sensitivity of optimized parameters in watershed models. *Water Resour. Res.* 14, 299–303. doi:10.1029/wr014i002p00299

Messalas, A., Kanellopoulos, Y., and Makris, C. (2019). "Model-agnostic interpretability with SHAPley values," in *2019 10th international conference on information, intelligence, systems and applications (IISA)*, 1–7.

Misirli, F., Gupta, H. V., Sorooshian, S., and Thiemann, M. (2003). Bayesian recursive estimation of parameter and output uncertainty for watershed models. *Calibration Watershed Models, Water Sci. Appl. Ser* 6, 113–124.

Model Analysis ToolKit (2021). *Python toolkit for model analysis*. Available at: http://dharp.github.io/matk/ (Accessed on 07, 202121).

Moore, R. B., and Dewald, T. G. (2016). The road to NHDPlus-advancements in digital stream networks and associated catchments. *J. Am. Water Resour. Assoc.* 52, 890–900. doi:10.1111/1752-1688.12389

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., and Veith, T. L. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Trans. ASABE* 50, 885–900. doi:10.13031/2013.23153

Mudunuru, M. K., Cromwell, E. L. D., Wang, H., and Chen, X. (2022). Deep learning to estimate permeability using geophysical data. *Adv. Water Resour.* 167, 104272. doi:10.1016/j.advwatres.2022.104272

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *J. Math. Psychol.* 47, 90–100. doi:10.1016/s0022-2496(02)00028-7

Nakshatrala, K. B., and Joshaghani, M. S. (2019). On interface conditions for flows in coupled free-porous media. *Transp. Porous Media* 130, 577–609. doi:10.1007/s11242-019-01326-7

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., et al. (2021). What role does hydrological science play in the age of machine learning? *Water Resour. Res.* 57, e2020WR028091. doi:10.1029/2020wr028091

Neitsch, S. L., Arnold, J. G., Kiniry, J. R., and Williams, J. R. (2011). *Soil & water assessment tool theoretical documentation, version 2009, Grassland, soil and water research laboratory-agricultural research service*. Temple, TX: Blackland Research Center-Texas AgriLife Research.

NERSC (2021). *National energy research scientific computing center*. Available at: https://www.nersc.gov/ (Accessed on 07, 202121).

Nott, D. J., Marshall, L., and Brown, J. (2012). Generalized likelihood uncertainty estimation (GLUE) and approximate Bayesian computation: What's the connection? *Water Resour. Res.* 48. doi:10.1029/2011wr011128

OLCF (2021). *Oak Ridge leadership computing facility*. Available at: https://www.olcf.ornl.gov/ (Accessed on 07, 202121).

Oruche, R., Egede, L., Baker, T., and O'Donncha, F. (2021). *Transfer learning to improve streamflow forecasts in data sparse regions. arXiv preprint arXiv:2112.03088.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.

Pool, S., Vis, M., and Seibert, J. (2018). Evaluating model performance: Towards a non-parametric variant of the Kling-Gupta efficiency. *Hydrological Sci. J.* 63, 1941–1953. doi:10.1080/02626667.2018.1552002

PRISM (2021). *A high-resolution spatial climate data for the United States*. Available at: https://prism.oregonstate.edu/ (Accessed on 07, 202121).

Qiu, J., Yang, Q., Zhang, X., Huang, M., Adam, J. C., and Malek, K. (2019). Implications of water management representations for watershed hydrologic modeling in the Yakima River basin. *Hydrol. Earth Syst. Sci.* 23, 35–49. doi:10.5194/hess-23-35-2019

Rahmani, F., Lawson, K., Ouyang, W., Appling, A., Oliver, S., and Shen, C. (2021). Exploring the exceptional performance of a deep learning stream temperature model and the value of streamflow data. *Environ. Res. Lett.* 16, 024025. doi:10.1088/1748-9326/abd501

Read, J. S., Jia, X., Willard, J., Appling, A. P., Zwart, J. A., Oliver, S. K., et al. (2019). Process-guided deep learning predictions of lake water temperature. *Water Resour. Res.* 55, 9173–9190. doi:10.1029/2019wr024922

Rolnick, D., Veit, A., Belongie, S., and Shavit, N. (2017). *Deep learning is robust to massive label noise*. arXiv preprint arXiv:1705.10694.

Rouholahnejad, E., Abbaspour, K. C., Vejdani, M., Srinivasan, R., Schulin, R., and Lehmann, A. (2012). A parallelization framework for calibration of hydrological models. *Environ. Model. Softw.* 31, 28–36. doi:10.1016/j.envsoft.2011.12.001

Rudi, J., Bessac, J., and Lenzi, A. (2020). *Parameter estimation with dense and convolutional neural networks applied to the FitzHugh-Nagumo ODE*. arXiv preprint arXiv:2012.06691.

Sadeghi, M., Asanjan, A. A., Faridzad, M., Nguyen, P., Hsu, K., Sorooshian, S., et al. (2019). PERSIANN-CNN: Precipitation estimation from remotely sensed information using artificial neural networks–convolutional neural networks. *J. Hydrometeorol.* 20, 2273–2289. doi:10.1175/jhm-d-19-0110.1

Samimi, M., Mirchi, A., Moriasi, D., Ahn, S., Alian, S., Taghvaeian, S., et al. (2020). Modeling arid/semi-arid irrigated agricultural watersheds with SWAT: Applications, challenges, and solution strategies. *J. Hydrology* 590, 125418. doi:10.1016/j.jhydrol.2020.125418

Sampson, K., and Gochis, D. (2018). *RF Hydro GIS pre-processing tools, version 5.0, documentation*. Boulder, CO: National Center for Atmospheric Research, Research Applications Laboratory.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Netw.* 61, 85–117. doi:10.1016/j.neunet.2014.09.003

Schwarz, G. E., and Alexander, R. B. (1995). State soil geographic (STATSGO) data base for the conterminous United States. *Tech. Rep.*

Sergeev, A., and Del Balso, M. (2018). *Horovod: Fast and easy distributed deep learning in Tensorflow*. arXiv preprint: 1802.05799.

Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resour. Res.* 54, 8558–8593. doi:10.1029/2018wr022643

Singh, V. P., and Frevert, D. K. (2003). "Watershed modeling," in *World water & environmental resources congress 2003*, 1–37.

Singh, V. P., and Frevert, D. K. (2010). *Watershed models*. Boca Raton, FL: CRC Press.

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

Sobol', I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput. Math. Math. Phys.* 7, 86–112. doi:10.1016/0041-5553(67)90144-9

Song, D. H., and Tartakovsky, D. M. (2021). *Transfer learning on multi-fidelity data*. arXiv preprint arXiv:2105.00856.

Sundararajan, M., Taly, A., and Yan, Q. (2017). "Axiomatic attribution for deep networks," in *International conference on machine learning*, 3319–3328.

Tague, C. L., and Band, L. E. (2004). RHESSys: Regional Hydro-Ecologic Simulation System-An object-oriented approach to spatially distributed modeling of carbon, water, and nutrient cycling. *Earth Interact.* 8, 1–42. doi:10.1175/1087-3562(2004)8<1:rrhsso>2.0.co;2

Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. Philadelphia, PA: SIAM.

Thiemann, M., Trosset, M., Gupta, H. V., and Sorooshian, S. (2001). Bayesian recursive parameter estimation for hydrologic models. *Water Resour. Res.* 37, 2521–2535. doi:10.1029/2000wr900405

Tolson, B. A., and Shoemaker, C. A. (2007). Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resour. Res.* 43. doi:10.1029/2005wr004723

Tsai, W.-P., Feng, D., Pan, M., Beck, H., Lawson, K., Yang, Y., et al. (2021). From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling. *Nat. Commun.* 12, 5988–6013. doi:10.1038/s41467-021-26107-z

Van Leeuwen, P. J., and Evensen, G. (1996). Data assimilation and inverse methods in terms of a probabilistic formulation. *Mon. Weather Rev.* 124, 2898–2913. doi:10.1175/1520-0493(1996)124<2898:daaimi>2.0.co;2

Van, S. P., Le, H. M., Thanh, D. V., Dang, T. D., Loc, H. H., and Anh, D. T. (2020). Deep learning convolutional neural network in rainfall–runoff modelling. *J. Hydroinformatics* 22, 541–561. doi:10.2166/hydro.2020.095

Westerberg, I. K., and McMillan, H. K. (2015). Uncertainty in hydrological signatures. *Hydrol. Earth Syst. Sci.* 19, 3951–3968. doi:10.5194/hess-19-3951-2015

Westerberg, I. K., Wagener, T., Coxon, G., McMillan, H. K., Castellarin, A., Montanari, A., et al. (2016). Uncertainty in hydrological signatures for gauged and ungauged catchments. *Water Resour. Res.* 52, 1847–1865. doi:10.1002/2015wr017635

Willard, J. D., Read, J. S., Appling, A. P., Oliver, S. K., Jia, X., and Kumar, V. (2022). *Predicting water temperature dynamics of unmonitored lakes with meta transfer learning*. Malden, MA: John Wiley & Sons Inc. e2021WR029579.

Wu, R., Chen, X., Hammond, G. E., Bisht, G., Song, X., Huang, M., et al. (2021). *Coupling surface flow with high-performance subsurface reactive flow and transport code PFLOTRAN*, 137. Environmental Modelling & Software.

Zhang, D., Chen, X., Yao, H., and James, A. (2016). Moving SWAT model calibration and uncertainty analysis to an enterprise Hadoop-based cloud. *Environ. Model. Softw.* 84, 140–148. doi:10.1016/j.envsoft.2016.06.024

Zhang, X., Srinivasan, R., and Van Liew, M. (2009). Approximating SWAT model using artificial neural network and support vector machine. *JAWRA J. Am. Water Resour. Assoc.* 45, 460–474. doi:10.1111/j.1752-1688.2009.00302.x

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., et al. (2020). A comprehensive survey on transfer learning. *Proc. IEEE* 109, 43–76. doi:10.1109/jproc.2020.3004555

Nomenclature

ARW American River Watershed

ATS Advanced Terrestrial Simulator

BNN Bayesian Neural Networks

CN Curve Number

CNN Convolutional Neural Network

DART The Data Assimilation Research Testbed

DHSVM The Distributed Hydrology Soil Vegetation Model

DDS Dynamically Dimensioned Search

DEM Digital Elevation Model

DNN Deep Neural Network

DL Deep Learning

ET Evapotranspiration

FAIR Findable Accessible Interoperable and Reusable

GIS Geographic Information System

GLUE Generalized Likelihood Uncertainty Estimation

GSA Global Sensitivity Analysis

GPU Graphical Processing Unit

HRU Hydrologic Response Unit

HSPF Hydrological Simulation Program-Fortran

MADS Model Analysis & Decision Support

KGE Kling-Gupta Efficiency

MATK Model Analysis ToolKit

mKGE Modified Kling-Gupta Efficiency

MI Mutual Information

ModEx Model-Experimentation

MTL Multi-Task Learning

NLCD National Land cover Database

NHDPlus National Hydrography Dataset Plus

npKGE Non-Parametric Kling-Gupta Efficiency

NSE Nash-Sutcliffe efficiency

logNSE Logarithmic Nash-Sutcliffe Efficiency

NWM The National Water Model

PET Potential Evapotranspiration

PEST Parameter Estimation Software

PRISM Parameter Elevation Regression on Independent Slopes Model

PRMS Precipitation Runoff Modeling System

RHESSys Regional Hydro-Ecologic Simulation System

SCE-UA Shuffled Complex Evolution Method developed at The University of Arizona

SWAT Soil and Water Assessment Tool

SWAT-CUP SWAT Calibration and Uncertainty Programs

SNOTEL Snow Telemetry

STATSGO Soil Maps for the State Soil Geographic

STL Single-Task Learning

TPU Tensor Processing Unit

USGS United States Geological Survey

WRF-Hydro The Weather Research and Forecasting Model Hydrological Modeling System

VIC The Variable Infiltration Capacity model

XAI Explainable AI

YRB Yakima River Basin