



OPEN

Decentralized digital twins of complex dynamical systems

Omer San^{1,2✉}, Suraj Pawar¹ & Adil Rasheed^{3,4}

In this article, we introduce a decentralized digital twin (DDT) modeling framework and its potential applications in computational science and engineering. The DDT methodology is based on the idea of federated learning, a subfield of machine learning that promotes knowledge exchange without disclosing actual data. Clients can learn an aggregated model cooperatively using this method while maintaining complete client-specific training data. We use a variety of dynamical systems, which are frequently used as prototypes for simulating complex transport processes in spatiotemporal systems, to show the viability of the DDT framework. Our findings suggest that constructing highly accurate decentralized digital twins in complex nonlinear spatiotemporal systems may be made possible by federated machine learning.

A recent multifaceted trend in science, engineering, and technology, both in academia and industry, is the digital transformation^{1,2}. The transformation has caught pace owing to cost-effective sensors, machine learning and artificial intelligence breakthroughs, improved computational infrastructure, and readily available course materials. However, this growth has resulted in several major challenges. First, the data exhibit a high correlation in many systems, which implies that more data does not necessarily translate into more information. Second, it is computationally demanding to train an end-to-end data-driven machine learning model that can be trustworthily used in future predictions. Third, there is a growing interest in the legal and regulatory landscape relevant to data sharing, security, privacy, and intellectual property rights (IPRs).

An outcome of rapid digitalization is the emergence of technologies such as digital twins. A digital twin can be defined as an evolving computational model of an asset or process^{3–5}. The significance of digital privacy, primarily focused on safeguarding individual data privacy during statistical analysis, becomes increasingly important alongside the development of digital twin technology, which is primarily concerned with creating virtual replicas of physical systems for analysis and optimization^{6–17}. Digital twins interact with physical assets in both ways: by controlling real assets and predicting their future states, and by calibrating models using data from the physical asset. The digital twin concept has gained a multibillion dollar market capitalization value for years to come, since it involves stakeholders from asset creation through decommissioning stages¹⁸. Numerous disruptive concepts and start-ups have been continuously evolving to bring more value and impact to society. According to a recent report by Research Dive¹⁹, the global digital twin market is predicted to grow at a compound annual growth rate of 40%, thereby garnering \$125 billion by 2030. In particular, the availability of highly modular open-source libraries such as TensorFlow, PyTorch, and Theano, and the flexibility of cross-platforms such as Unity and Unreal Engine are lowering the barriers for many different potential use cases from precision agriculture²⁰ to precision medicine²¹ and beyond. This open-source ecosystem has allowed scientists and engineers to expose themselves faster to the state-of-the-art digital twin technologies with cross-functionalities.

One of the key enablers of digital twins is modeling, which can either be first-principle physics-based modeling (PBM) or data-driven modeling (DDM) each with weaknesses that limits their usage in a digital twin context. To address the weaknesses, more recently, hybrid analysis and modeling (HAM)⁴ which is a fusion between PBM and DDM is rapidly evolving (see, e.g., see also^{22–31}). Of the three modeling approaches, DDM and HAM thrive on the availability of a large amount of high quality data. The data requirement in the context of digital twins poses several challenges, especially as one moves towards creating digital twins of increasingly complex systems. We enumerate the challenges that are most relevant in the context of the current work:

- Any reasonably complex asset of a physical system or a process consists of multiple sub-components operated and maintained by different players who might have competing interests making data sharing difficult.

¹School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74078, USA. ²Department of Mechanical, Aerospace and Biomedical Engineering, University of Tennessee, Knoxville, TN 37996, USA. ³Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7465 Trondheim, Norway. ⁴Department of Mathematics and Cybernetics, SINTEF Digital, 7034 Trondheim, Norway. ✉email: osan@utk.edu

- Even when the IPRs involving data are sufficiently resolved, the shared big data characterized by 5Vs (large volume, velocity, veracity, variety, value) from different components/vendors can become overwhelmingly large that might require an exorbitant amount of compute power in one place to generate knowledge.
- Furthermore, the analysis of a large variety of big data in a centralized approach requires a wide array of expertise in one place, which will be difficult to achieve, resulting in sub-par analysis.
- Lastly, the limited bandwidth of data transmission in a centralized data analysis approach limits the amount of shared data.

To that end, the concept of federated learning can be utilized. Federated learning facilitates collaborative model training on decentralized devices (clients) housing local data, allowing updates to the global model without centralizing raw data, thereby preserving data privacy across distributed entities. Rich applications for training a machine learning model across numerous dispersed edge devices are emerging from this idea, and they do so without transferring local data samples across servers^{32–35}. Although such privacy preserving decentralized and collaborative machine learning applications become more of a main stream approach in advertising, financial and many other industries with personalized dominance, it is relatively uncharted in advancing functionalities and providing technical support to forge a cross-domain, cross-data and cross-enterprise digital twin ecosystem that needs to be compatible with the diversity of scales present in data. Federated machine learning algorithms have been successfully used for training machine learning models in broader applications ranging from service recommendation to wireless communication. However, the multifidelity, multiphysics and multiscale nature of the many engineering and transport systems poses additional challenges because the accurate representation of each fidelity does not guarantee an improved metrics at the system level, partially because of nonlinear interactions across scales and also due to errors which may propagate and accumulate³⁶.

In this study, we aim to extend the prospects of this concept to complex dynamical systems in order to provide building blocks (from a modeling perspective) for next generation decentralized solvers. With this in mind, our chief motivation is to communicate to a wide variety of audiences on how federated machine learning can address many of the issues raised above. Consequently, this is more of a concept paper, and we omit providing rigorous derivations of the utilized algorithms. Consistently, several relatively simple (but representative) problems have been chosen to demonstrate the framework. The plain vanilla versions of learning models have been utilized without any particular efforts dedicated to rigorous hyperparameter tuning and model calibration. Rather, our focus is laser sharp on highlighting the exploitation of the benefits of federated machine learning in the context of digital twin.

Federated learning in the context of digital twins

Digital twins

Digital twin, which is a virtual version of a physical asset that is made possible by data and simulators for improved decision-making and real-time prediction, optimization, monitoring, and control⁵, is a technology penetrating every domain. Figure 1 describes the concept of a digital twin utilizing a decentralized modeling for physical realism. On the top right side of the figure, we have the physical asset we want to represent by a digital

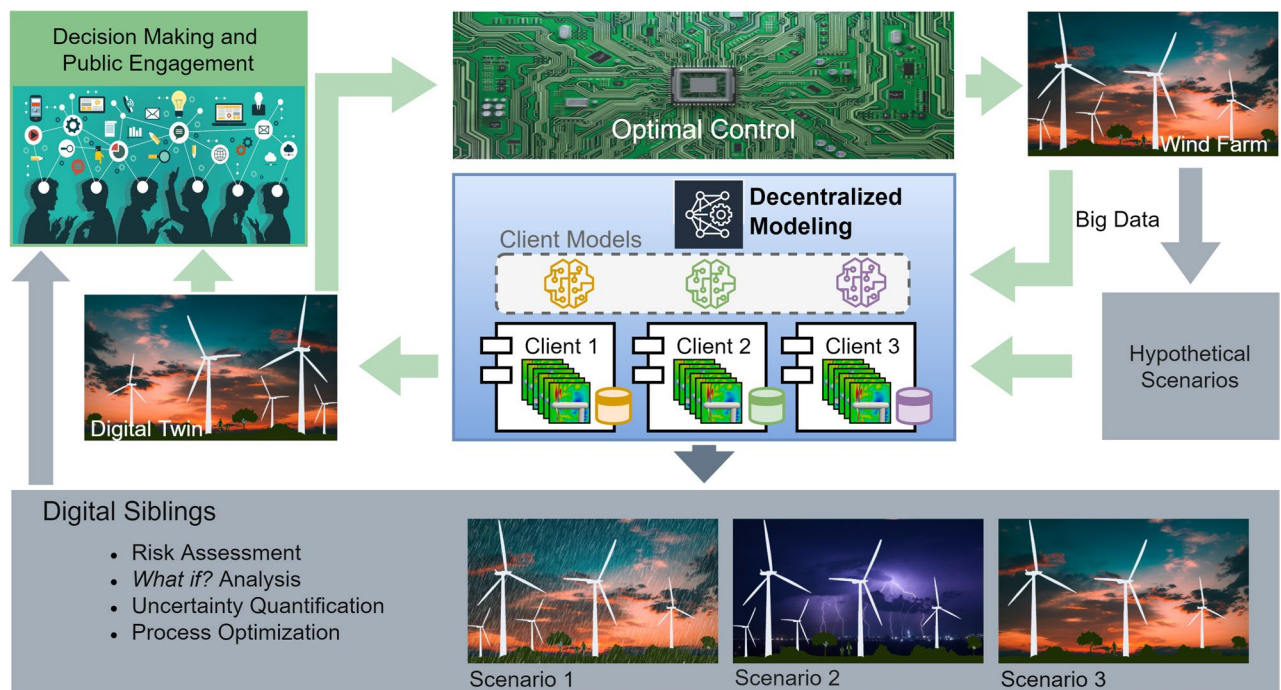


Figure 1. An overview of a decentralized digital twin concept for wind farm applications.

twin. The physical asset is often equipped with a diverse class of sensors that give big data in real-time. This data has a very coarse spatio-temporal resolution and does not describe the future state of the asset. Therefore, to complement the measurement data, models are utilized to bring physical realism into the digital representation of the asset. Provided that the same information can be obtained from the digital twin as from the physical asset, we can utilize the digital twin for informed decision-making and optimal control of the asset. However, one might be interested in risk assessment, what-if? analysis, uncertainty quantification, and process optimization. These can be realized by running the digital twin in an offline setting for scenario analysis. The concept is then known as a digital sibling.

Additionally, the digital twin predictions can be archived during the asset's lifetime and can then be used for designing the next generation of assets, in which case the concept is referred to as digital threads. Digital twins, based on their capability levels, can be categorized on a scale from 0 to 5 as standalone, descriptive, diagnostic, predictive, prescriptive, and autonomous. It is evident that modeling plays a vital role in enhancing the capability of digital twins. As pointed out in a recent survey⁴, any modeling approach should at least be generalizable, trustworthy, computationally efficient, accurate, and self-adapting.

Since a digital twin requires a real-time exchange of data between the physical asset and its digital representation, data privacy and security objectives become an increasingly important topic. Due to the participation of multiple stakeholders all the way from the development, through operation to the decommissioning phase of physical assets, the gathered data has to be secure both in terms of validity and privacy. Many digital twin systems and applications might involve different original equipment manufacturers, and they might want to keep their proprietary rights to secure their designs.

The predictive digital twins are now coined with precision sciences, such as precision agriculture²⁰, precision medicine²¹ as well as precision meteorology³⁷, in emphasizing their improved and localized forecasting capabilities. For example, we emphasize the penetration of data-driven modeling in precision meteorology, which often refers to accurate microscale weather forecasting that is of great importance in enabling solutions to water, food, energy, and climate challenges in the coming century³⁸. Therefore, an increase in parameterization capabilities and characteristics along with enhanced in situ observations within the atmospheric boundary layer becomes a key concept to improve the accuracy of such microscale weather forecasting models³⁹.

Our chief motivation in this study relies on the fact that a greater than ever penetration of sensors and smart devices (e.g., smart weather stations, smartphones, and smartwatches) has remained an unexplored technology in many spatiotemporal extended systems, and crowdsourcing data-driven modeling could be a key enabler toward their emerging digital twin applications. For instance, there will be more than 7 billion smartphones worldwide by 2025⁴⁰. Compared to the few official meteorological stations distributed throughout the world, this number is large. While the use of data from a small number of edge devices may not result in reliable forecasts, processing data from several connected, smart devices that have sensors may revolutionize weather monitoring and forecasting. Numerous applications of smartphones toward citizen-centered monitoring, detection and warning systems have been discussed in a recent overview⁴¹ including examples from earthquake predictions to traffic, pavement and structural engineering projects. All of these case studies support the utilization of smartphones as cost-effective instruments for collecting real-time data. Moreover, as discussed in a recent report⁴², the Weather Company uses information from over 250,000 personal weather stations. Furthermore, Chapman et al.⁴³ provided their insights on how using Netatmo weather sensors and the crowdsourcing data-driven modeling paradigm might advance the field of meteorology. The security and privacy risks of such linked and smart weather stations have also been highlighted as more focus switches to the internet of things (IoT) devices⁴⁴. Additionally, we point readers to recent discussions on IoT uses for meteorological purposes and low-cost design of weather stations^{45–48}. Furthermore, it is noteworthy that there is a contemporary inclination towards the utilization of federated large language models⁴⁹. This trend is propelled by the fact that data generation is primarily driven by end users, and when coupled with advancements in wearable device technology⁵⁰, federated models might exhibit a heightened effectiveness, particularly when harnessed in the context of generative models, for the purpose of adapting to screenless computing devices.

Federated learning

Federated machine learning is the collective training of machine learning models without the actual data being shared. We follow the pioneering work in federated learning⁵¹, which provides a federated averaging technique in which clients cooperatively train a shared model. The federated learning strategy is contrasted with the centralized approach in Fig. 2. Typically, the local dataset is moved from clients to a central server and the model is trained using data that is saved there when utilizing the centralized approach. However, the local dataset is never sent from clients to a server in federated learning. Instead, based on the local dataset, each client iteratively computes an update to the global model kept by the server, and only weights of the model are sent in each round. The federated averaging approach makes the assumption that a synchronous updating scheme is used in rounds of communications and that there is a fixed set of K clients with a fixed local dataset. The global state of the model, or the current model parameters, are sent by the central server to each client at the start of each communication round. Based on the global state and local dataset, each client computes the update to the global model, which is then communicated to a server. Following then, the server updates the model's overall state using the local updates that it has received from each client, and the cycle repeats. The federated averaging algorithm's objective function can be expressed as follows

$$f(w) = \sum_{k=1}^K \frac{n_k}{n} F_k(w), \quad (1)$$

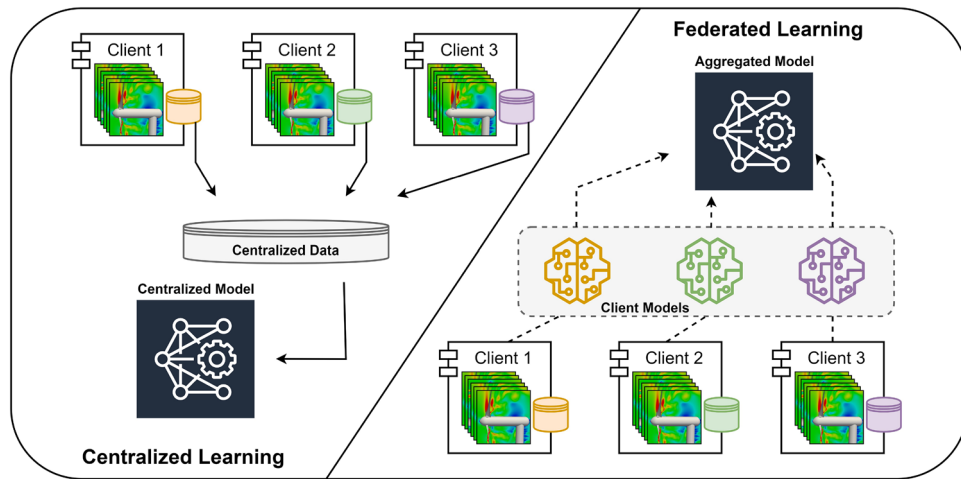


Figure 2. Overview and schematic illustrations of the centralized and federated machine learning approaches.

where

$$F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w). \tag{2}$$

Here, w denotes the parameters of the trainable model, n_k denotes the cardinality of \mathcal{P}_k , $f_i(w) = l(x_i, y_i; w)$ denotes the loss of the prediction on sample (x_i, y_i) , and \mathcal{P}_k denotes the data on the k th client. Any machine learning algorithm can be used with the aforementioned aggregation technique. We emphasize that the methodology we employ in our study^{51, 52} only weights edge devices proportionally according to the data they possess. These restrictions can be mitigated by more sophisticated methods^{53–57}, but such methods are outside the purview of this paper.

In accordance with the provided pseudocode from prior research⁵², the primary server initializes the process by transmitting the global model state (i.e., model parameters) to all participating clients at the outset of each communication round. Subsequently, the central server accumulates these gradients and applies an update rule to the received data, producing the next iteration of model parameters through a weighted-average aggregation of local gradients. These updated parameters are then dispatched by the central server to the active clients. In essence, each client utilizes its local data to execute a single gradient descent step on the current model, after which the server computes a weighted average of the resulting models. This process is repeated for a specified number of gradient descent steps conducted by each client on their local dataset before transmitting their updates to the server. Consequently, varying federated algorithms can be formulated based on the number of training cycles executed by clients and the selection criteria for clients participating in each parameter update round.

Demonstration cases

To demonstrate the value of the federated learning we chose the following frameworks and testbeds.

Federated reduced order modeling framework

The one-dimensional Burgers equation, a common example of a nonlinear advection-diffusion problem, serves as our first test case for illustrating the DDT framework.

The viscous Burgers equation can be written as follows

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, 1], \tag{3}$$

where $\mathbf{u}(x) \in \mathbb{R}^{N_x}$ is the velocity field. We can write an analytical solution of the Burgers equation that satisfies Eq. 3 as follows

$$u(x, t) = \frac{\frac{x}{t+1}}{1 + \sqrt{\frac{t+1}{t_0} \exp\left(\frac{x^2}{4\nu(t+1)}\right)}}, \tag{4}$$

where $t_0 = \exp(1/8\nu)$. We parameterize the Burgers equation using the time t and viscosity ν as the parameter space, i.e., $\boldsymbol{\mu} = (t, \nu)^T$. The first step of the reduced order modeling (ROM) is to generate the data. We generate the data for different values of parameters. Specifically, the parameter t is varied between 0 and 2, and the parameter ν lies between 0.001 to 0.01. The matrix $\mathbf{A} \in \mathbb{R}^{N_x \times N}$, whose columns are the \mathbf{u}_n corresponding to $\boldsymbol{\mu}_n$, is formed after the data is generated. The matrix is then subjected to singular value decomposition

$$A = W \Sigma V^T = \sum_{k=1}^N \sigma_k \mathbf{w}_k \mathbf{v}_k^T, \tag{5}$$

where W is an $N_x \times N$ matrix with orthonormal columns, \mathbf{w}_k , V , which is an $N \times N$ square matrix with orthonormal columns, \mathbf{v}_k , and Σ is also an $N \times N$ matrix with non-negative diagonal entries, often known as singular values, arranged so that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_N \geq 0$. The proper orthogonal decomposition (POD) modes are represented by the vectors \mathbf{w}_k , and the set of POD basis functions is represented by $\Phi = \{\phi_k\}_{k=1}^R$ ⁵⁸.

The following is then a representation of the approximate velocity field using POD modes

$$\mathbf{u}(\mathbf{x}, \boldsymbol{\mu}) = \sum_{k=1}^R \alpha_k(\boldsymbol{\mu}) \phi_k(\mathbf{x}), \tag{6}$$

where R is the number of modes that are retained in our model, and $\alpha_k(\boldsymbol{\mu})$ denotes the k th modal coefficient.

The relative information content (RIC) can be used to determine how much energy POD modes maintain. We determine R from the RIC index given by

$$\text{RIC}(R) = \frac{\sum_{j=1}^R \sigma_j^2}{\sum_{j=1}^N \sigma_j^2}, \tag{7}$$

where we set $\text{RIC}(R) = 99.99\%$.

We first project snapshot data to POD modes (i.e., forward POD transformation). The neural network is then trained to learn the relationship between the parameters of the Burgers equation to the POD modal coefficients. Once the neural network is trained, we can predict the POD modal coefficients for a new set of parameters (i.e., inverse POD transform), and the velocity field can be reconstructed using Eq. 6. In the case of the federated ROM framework, the low-dimensional data is assumed to be distributed across clients and the training is done using the federated averaging algorithm⁵¹. The complete federated ROM framework is illustrated in Fig. 3 where clients collaboratively learn an aggregated model while keeping all the training data on their respective client. The machine learning model is a feed-forward neural network with four hidden layers and forty neurons per hidden layer. We use the rectified linear unit (ReLU) activation function to introduce nonlinearity and the linear activation function is used at the output layer.

Federated autoencoder framework

Our second test case is the decentralized autoencoder for the nonlinear dimensionality reduction of the NOAA OI SST V2 analysis dataset. Autoencoders are widely used in unsupervised machine learning tasks. Closely following⁵², we use an encoder network to map an input to a low-dimensional latent space, and a decoder network to translate latent variables back into the original inputs. This dataset has been created utilizing in-situ (ship and buoy) and satellite sea surface temperature (SST) measurements along with sea-ice cover simulations of SSTs. The

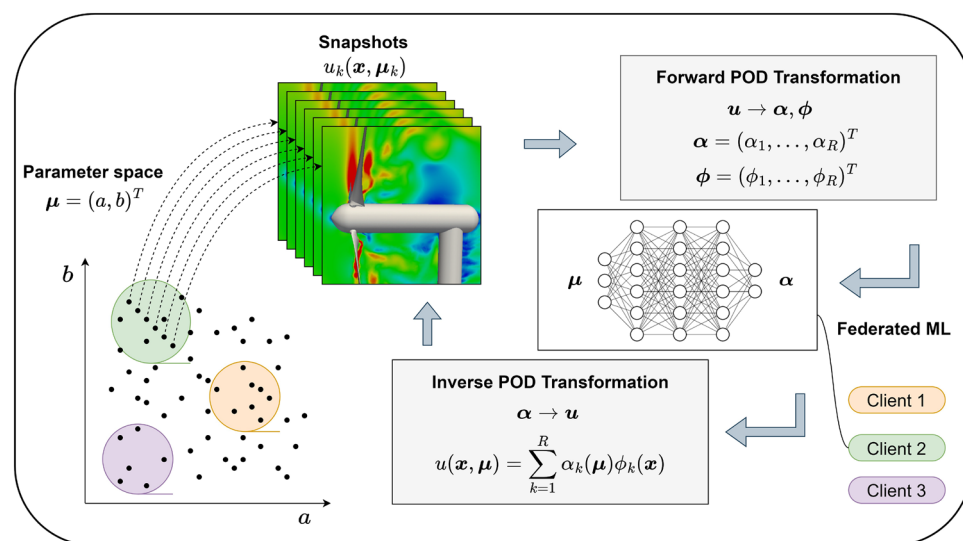


Figure 3. Federated reduced order modeling framework. This approach enables clients to collaboratively learn an aggregated model while keeping all the training data on each client. In other words, it encourages knowledge sharing without sharing the actual data. It deals with training a model across multiple decentralized edge devices (or servers) without exchanging local data samples between them.

satellite data is corrected for biases^{59,60} before the analysis is computed. This dataset is composed of the weekly average sea surface temperature on a 1° latitude \times 1° longitude global grid (180×360).

Despite seasonal fluctuations giving the SST dataset a strong periodic structure, complicated ocean dynamics produce interesting flow physics in this dataset. This dataset has been used in a variety of recent works, including multi-resolution dynamic mode decomposition⁶¹, flow reconstruction⁶², and geophysical emulation⁶³.

More specifically, the autoencoder is utilized for nonlinear dimensionality reduction of the NOAA OI SST V2 dataset. The autoencoder is composed of the encoder function $\eta(w)$ which maps the high dimensional sea surface temperature field to low dimensional latent space and a decoder function $\xi(w)$ which maps the low dimensional latent space to the same sea surface temperature field. Both encoder and decoder are parameterized by the weights w and these parameters are learned through training. We can represent the reconstruction of the temperature field with the autoencoder as follows

$$\eta(w), \xi(w) = \arg \max_{\eta(w), \xi(w)} \|\theta - (\eta(w) \circ \xi(w))\theta\|, \quad (8)$$

$$\text{Encoder} := \eta(w) : \theta \in \mathbb{R}^N \rightarrow z \in \mathbb{R}^R, \quad (9)$$

$$\text{Decoder} := \xi(w) : z \in \mathbb{R}^R \rightarrow \theta \in \mathbb{R}^N, \quad (10)$$

where θ is the sea surface temperature field, z represents the low dimensional latent space and R is the dimensionality of the latent space at the bottleneck layer of the autoencoder. Here, we train the autoencoder model using the data from October 1981 to July 2010 (1500 snapshots), and we compare the performance of the trained autoencoder model with the unseen data using the data collected after July 2010.

Federated physics-guided machine learning framework

In our final demonstration case, we devise a framework integrating physics-guided machine learning (PGML) within a federated environment to construct a DDT model, which was then applied to lift coefficient data from various NACA standard airfoils for comprehensive training and testing across diverse conditions. As elucidated by Robinson et al.⁶⁴ in their works, PGML operates as a HAM approach merging simplified theories to construct robust machine learning models, integrating this knowledge both during training and deployment phases. Consequently, the neural architecture, including hyperparameter optimization and the definition of nodes and layers, plays a pivotal role in PGML's efficacy, with a trade-off observed: overly narrow networks exhibit limited expressiveness impacting predictions, while overly complex architectures, coupled with insufficient training data, can lead to overfitting and reduced generalizability, as indicated by an escalating validation loss after several training iterations.

The concept of injecting data into machine learning models revolves around the convergence of data sources related to a physical system, encompassing data from high-fidelity numerical simulations, experiments, or various data acquisition methods such as in-situ sensors, satellite observations, and field measurements, all contributing to the understanding of the system's behavior as a unified reality. In this study, we extend our prior PGML model⁶⁵ into a federated architecture. In the current context, the concept of inductive bias holds paramount importance for efficient DTT modeling. Inductive bias encompasses a set of assumptions regarding the underlying phenomenon or data structure, typically derived from prior knowledge and experience, enabling effective adaptation to novel data. In the context of federated computing, we advocate the incorporation of inductive bias into the internal layers of neural operators. Broadly, every assumption and belief concerning the data represents an instance of inductive bias, which mitigates prediction uncertainty in ML models and facilitates early resolution of learning and memory issues. It is worth noting that inductive bias proves particularly reliable for overparameterized systems, rendering our approach a viable strategy for decentralized digital twins.

As shown in Fig. 4, each client possesses distinct airfoil geometries, and without sharing geometric specifics or relevant flow field data, our objective is to train a global model accommodating diverse local scenarios. Clients collaborate in learning an aggregated model while safeguarding the privacy of their individual training data. This decentralization of training data ensures data privacy, as raw data is not exchanged with the central aggregator. Clients utilize their own computational resources to perform local computations on their datasets, incorporating shared parameters based on the global model's state. Consequently, the sole information transmitted to the server comprises the model updates generated by the clients using their local data. Upon receiving these updates, the parameter server adjusts its global model by updating the weights learned from each client's local dataset through iterative iterations. In summary, our approach involves two critical phases: local updating with PGML network, where clients perform gradient descents to minimize the local loss function with respect to their data, and global aggregation, encompassing the collection of updated model parameters from multiple client devices, their consolidation, and the subsequent distribution of the aggregate parameters to clients for further training iterations. Utilizing the identical architecture as previously detailed in our earlier work⁶⁵, we abstain from reiterating the specifics of the PGML network in this context. Instead, we present this example as an illustrative case showcasing the seamless extension of a specific PGML architecture into its federated counterpart while preserving its inherent characteristics and feasibility.

Results

Federated reduced order modeling framework: Burgers system

In our demonstrations, we employ $K = 10$ clients for the federated learning, with each client model being trained for 5 local epoch with a batch size of 32. The centralized model is trained with a batch size of 320 to ensure fair

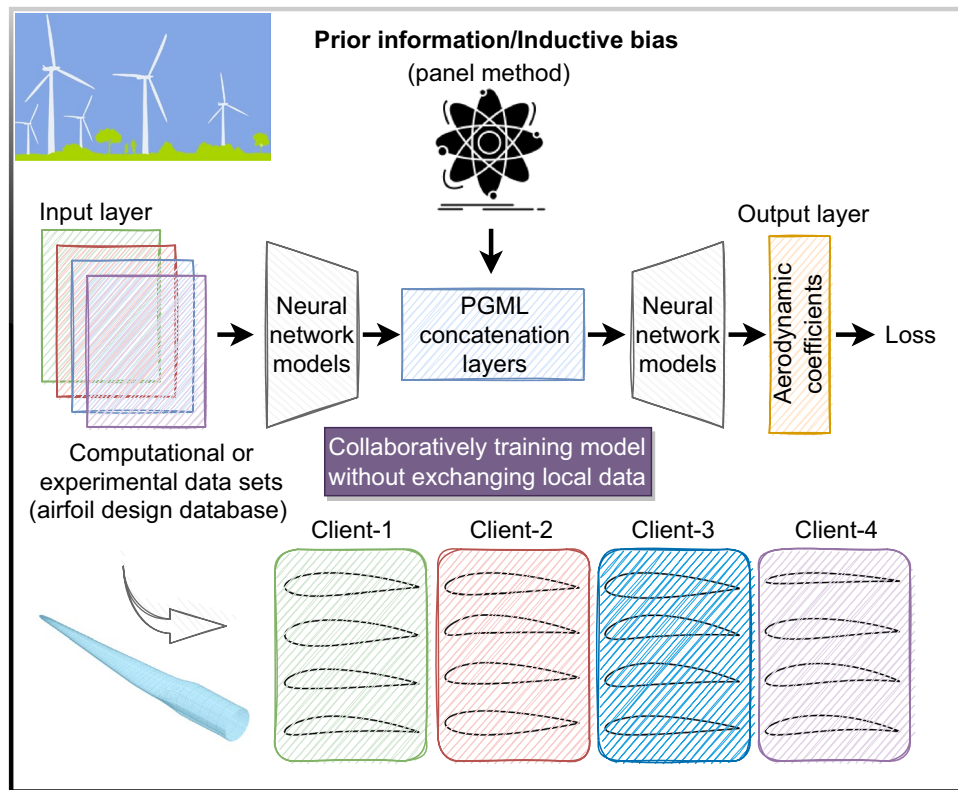


Figure 4. Illustration of the federated physics guided machine learning framework as a key enabler toward decentralized digital twins.

comparison. The mean squared error loss for the validation dataset during the training of centralized and federated ROM frameworks is shown in Fig. 5. The validation loss for both centralized and federated learning follows a similar trajectory during the training indicating that there is no loss in federated learning. Therefore, federated learning can provide us with a deep learning model with a similar level of accuracy as centralized learning without the need to share the data. The performance of the parametric ROM framework for $\mu = (0.02, 0.00475)^T$ is shown in Fig. 6. Compared against the FOM data, which is the analytical (true) solution in this example, we see that both centralized and federated ROMs are highly accurate in reconstructing the velocity field for the Burgers equation.

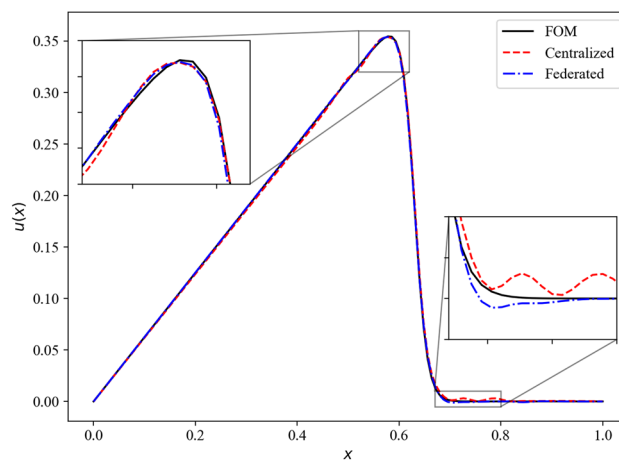


Figure 5. Validation loss during training of the centralized and federated neural network for the ROM framework of Burgers equation.

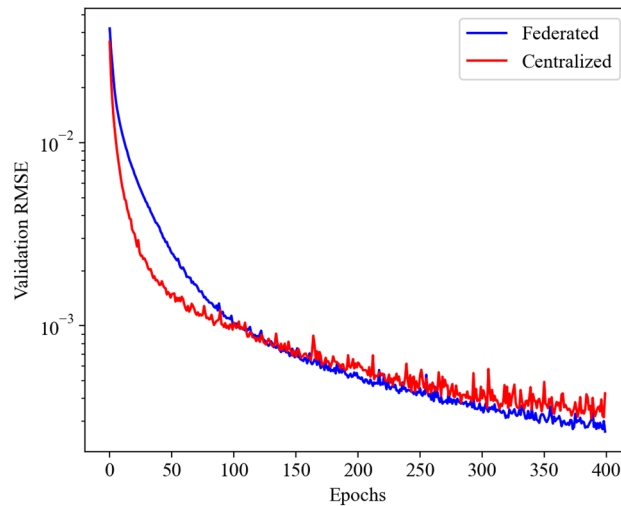


Figure 6. Reconstruction performance of the centralized and federated learning approaches for Burgers equation for $\mu = (0.02, 0.00475)^T$.

Federated autoencoder framework: NOAA optimum interpolation sea surface temperature dataset

First, we show the performance of centralized and decentralized or federated autoencoders for different dimensionality of the latent space at the bottleneck layer. The encoder is composed of three hidden layers with the number of neurons being {800, 400, 200} across the first, the second, and the third hidden layer, respectively. The decoder has the same architecture in reverse order, i.e., three hidden layers with {200, 400, 800} neurons. The number of neurons in the bottleneck layer is varied between $R = 10$ to $R = 25$ with an increment of 5. The encoder and decoder architecture remain the same for all the numerical experiments. In each case, we use $K = 10$ clients, with each client model trained for 1 local epoch with a batch size of 16. To guarantee fair comparison, the centralized model is trained with a batch size of 160. Figure 7 depicts the validation loss during both centralized and federated training of the autoencoder. The validation loss for federated learning is slightly higher compared to the centralized learning from the first epoch. This can be attributed to the fact that in federated averaging the weights of the global model are initialized by simply averaging the weights of the client model. The simple averaging of weights might not be the most effective strategy especially when the amount of the data is small for each client. Despite this, the validation loss for both centralized and federated learning follows a similar trajectory with the loss reduced by almost one order of magnitude from the initial loss. The reconstruction of the weekly average temperature field for the fourth week of March 2018 is shown in Fig. 8 for both centralized and federated autoencoder along with the full order model, i.e., the analysis.

Federated PGML framework: Aerodynamic characterization

In this demonstration, the model was specifically trained, tested, and validated using NACA 4-digit and 5-digit series data pertaining to variations in the lift coefficient for diverse airfoil types. In line with the PGML model⁵⁵, we extend its federated architecture to leverage pertinent physics-based characteristics from the panel method, enhancing the generalizability of data-driven models in a federated setting. For training purposes, numerical simulations are employed to generate data for a federated neural network involving $K = 10$ clients. The primary objective of this section is to compare federated and centralized approaches concerning prediction accuracy, both enriched with inductive bias and tested with distinct data sets from their training data. The training dataset encompasses lift coefficient data across various Reynolds numbers spanning from 1×10^6 to 4×10^6 and angles of attack ranging from -20 to $+20$, amounting to a total of 168 sets of two-dimensional airfoil geometries. This dataset covers NACA 4-digit, NACA210, NACA220, and NACA250 series airfoils, each with 201 points, featuring maximum thickness ratios ranging from 6% to 18% of the chord length. The testing dataset comprises NACA23012 and NACA23024 airfoil geometries, notably exceeding the maximum thickness ratios present in the training dataset and originating from a different NACA230 series not included in training.

Each client in the federated environment adopts a neural network architecture with four hidden layers, each containing 20 neurons. At the third hidden layer, latent variables and physical parameters, specifically the Reynolds number and angle of attack, are concatenated. By incorporating physical flow parameters, lift coefficient, and pressure drag coefficient predictions from the panel technique, we augment the latent variables at the third layer of the PGML model. The machine learning model consists of a feed-forward neural network with four hidden layers, each housing 20 neurons, and employs the ReLU activation function to introduce nonlinearity. To gauge predicted uncertainty, we employ an ensemble approach, combining predictions from multiple neural networks. These networks are trained with varying weight initializations and serve as an indicator of epistemic uncertainty. The ensemble of models is created by initializing each model's weights and biases using the Glorot

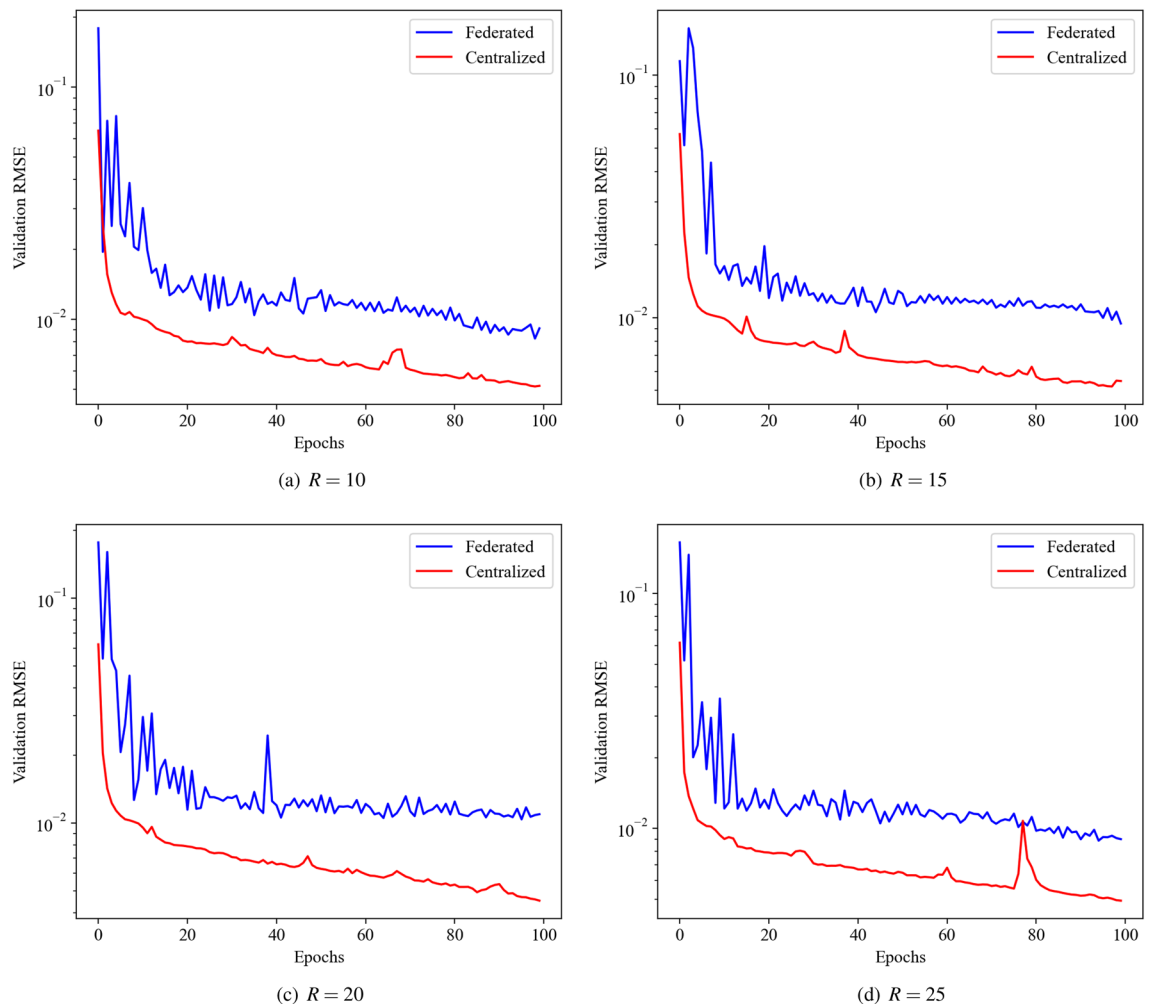


Figure 7. Validation loss during training of the centralized and federated neural network for the NOAA Optimum Interpolation Sea Surface Temperature Dataset.

uniform initializer and employing various random seed numbers to ensure unique parameter sets for each model. The ensemble collectively provides an estimate of the expected lift coefficient model uncertainty.

Figure 9 presents the lift coefficient predictions for NACA23012 and NACA23024 airfoils to illustrate the difference between standard ML and PGML. The PGML model provides more precise lift coefficient predictions with reduced uncertainty compared to the ML model for both airfoil types, particularly within an angle of attack range of -10° to $+10^\circ$. Higher uncertainty is expected outside this angle of attack range due to the successful application of the Hess-Smith panel method in analyzing inviscid flow for smaller angles of attack. It's worth noting that airfoil thickness in the training dataset does not exceed 18% of the chord length. The NACA23024 airfoil exhibits greater uncertainty in ML model predictions compared to the NACA23012 airfoil. The incorporation of physics-based information from the panel approach in the PGML model reduces this uncertainty. These findings highlight the advantages of enhancing neural network models with condensed theories and employing the PGML framework in physical systems.

Figure 10 displays the reconstruction RMSE for centralized and federated modeling approaches, both for 100 and 800 training epochs. Divergent convergence trajectory characteristics are observed in the context of centralized and federated ML and PGML methodologies as the number of training epochs increases. While both approaches exhibit error values converging to similar levels, the PGML method demonstrates slightly lower errors, particularly in extended training epochs. It is noteworthy that federated learning plateaus at a higher error level, as indicated by the initial rapid reduction in loss within the first 100 epochs, followed by a more gradual decrease compared to centralized learning. This sharper convergence of the centralized approach results in a lower RMSE in loss, contrary to the trend depicted in Fig. 6. Consequently, our investigation suggests that instead of making a blanket assertion that federated learning is as effective as centralized learning, it is imperative to acknowledge the disparity in their convergence trajectories. This divergence may be attributed to a combination of various factors, including hyperparameters, batch sizes, data distribution imbalances, and the non-monotonic characteristics inherent to neural network-based approaches. Furthermore, the centralized model shows noticeable error fluctuations, absent in the federated model, due to the noise reduction effect in federated learning. This arises from the averaging of model parameter values contributed by participating clients.

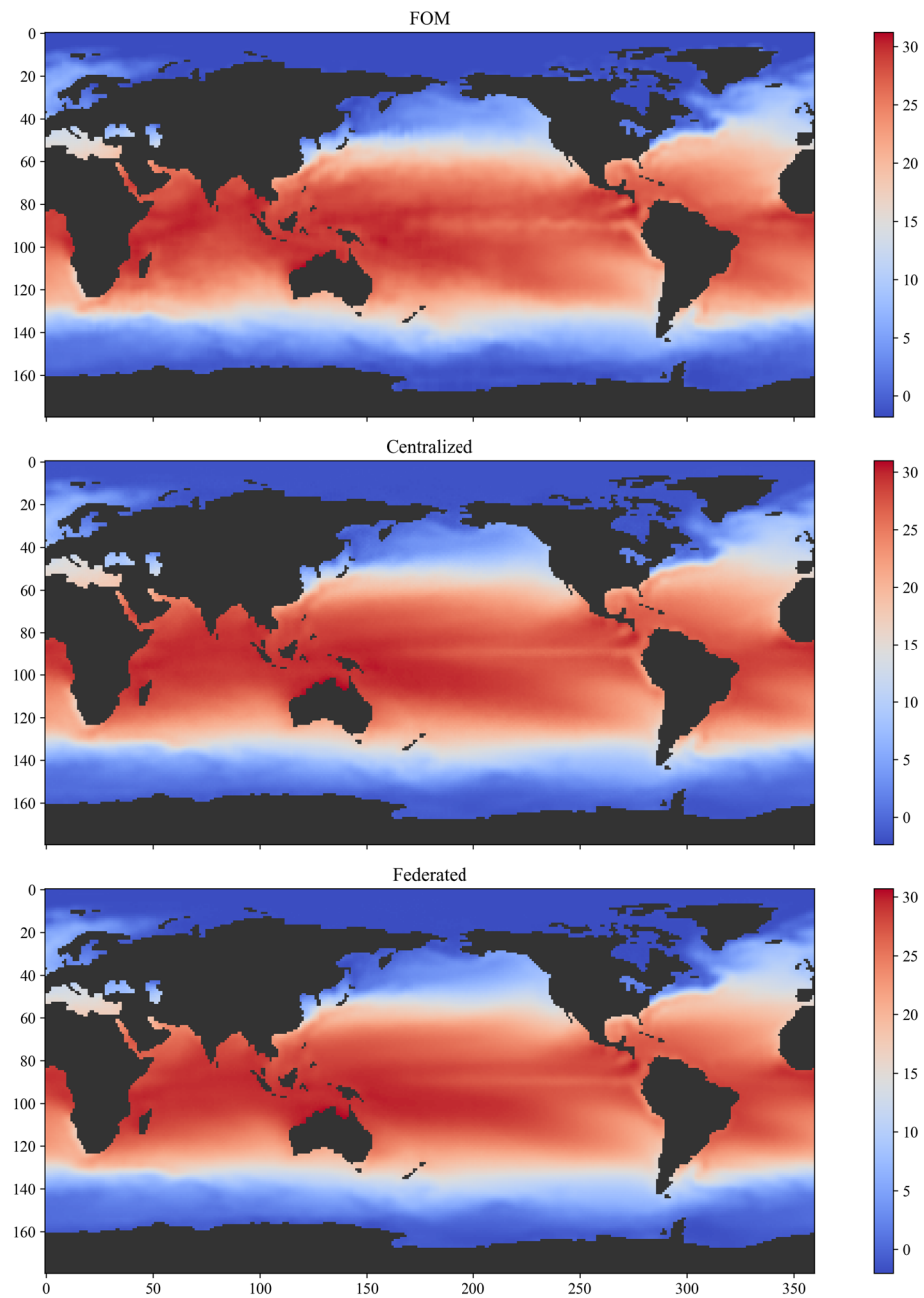


Figure 8. Weekly averaged temperature field in degrees Celsius for the fourth week of March 2018 along with reconstructed temperature field from the centralized and federated autoencoders.

Figure 11 investigates the models' performance concerning the angle of attack for both NACA23012 and NACA23024 inputs, using both 100 and 800 training epochs. Predictions by all models are more accurate within the angle of attack range of -10° to $+10^\circ$, thanks to the superior accuracy of the panel method in this range. Models trained with NACA23012 data generally outperform those trained with NACA23024 data, which requires extrapolation, leading to deviations from the reference. Despite this, federated models demonstrate comparable or even superior performance, especially when tested with data outside the training range. The 800-epoch comparison in Fig. 11 highlights the stability of models trained with NACA23012 data throughout the entire angle of attack spectrum. However, with longer training epochs, a noticeable difference emerges between federated and centralized methods for models using NACA23024 data. Federated learning performs proportionally with centralized methods, underscoring the noise-canceling and filtering benefits of the federated environment, as shown in Fig. 12.

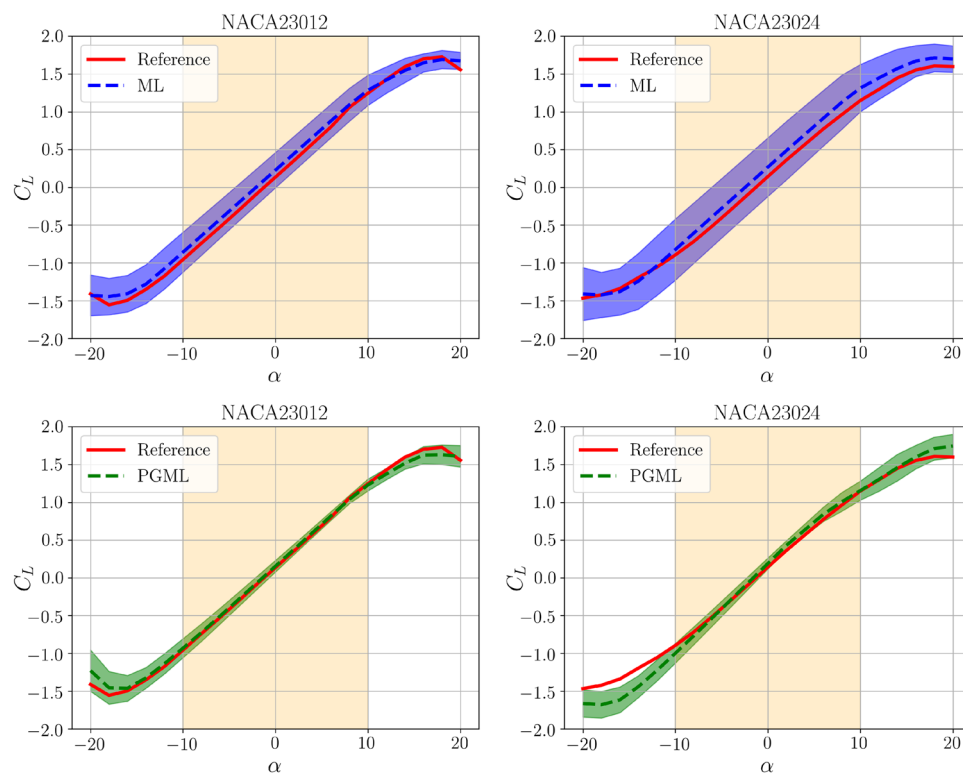


Figure 9. Uncertainty quantification of ML and PGML approaches using the centralized training.

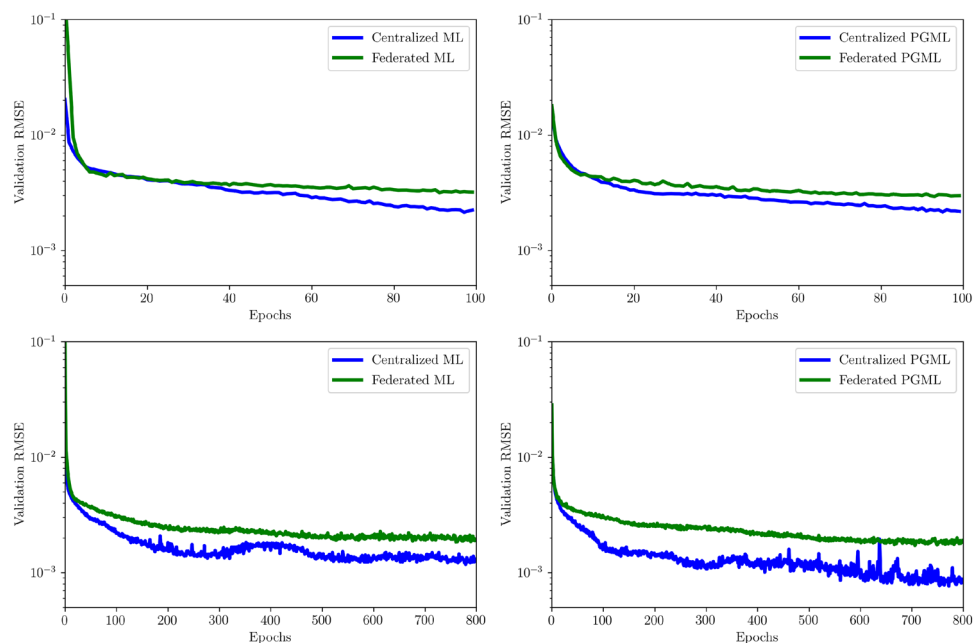


Figure 10. Validation loss of the proposed centralized and federated approaches. The first row shows the relaxation histories for the early truncation of 100 epochs, while the second row illustrates up to 800 epochs.

Discussion

In this work, we explore the potential of federated machine learning in the context of a digital twin operated by multiple vendors to model complex spatio-temporal dynamical systems. In particular, we investigated federated learning for physics-guided neural networks and nonlinear dimensionality reduction of dynamical systems. Federated learning makes it possible to train a model jointly while maintaining the decentralized nature of the

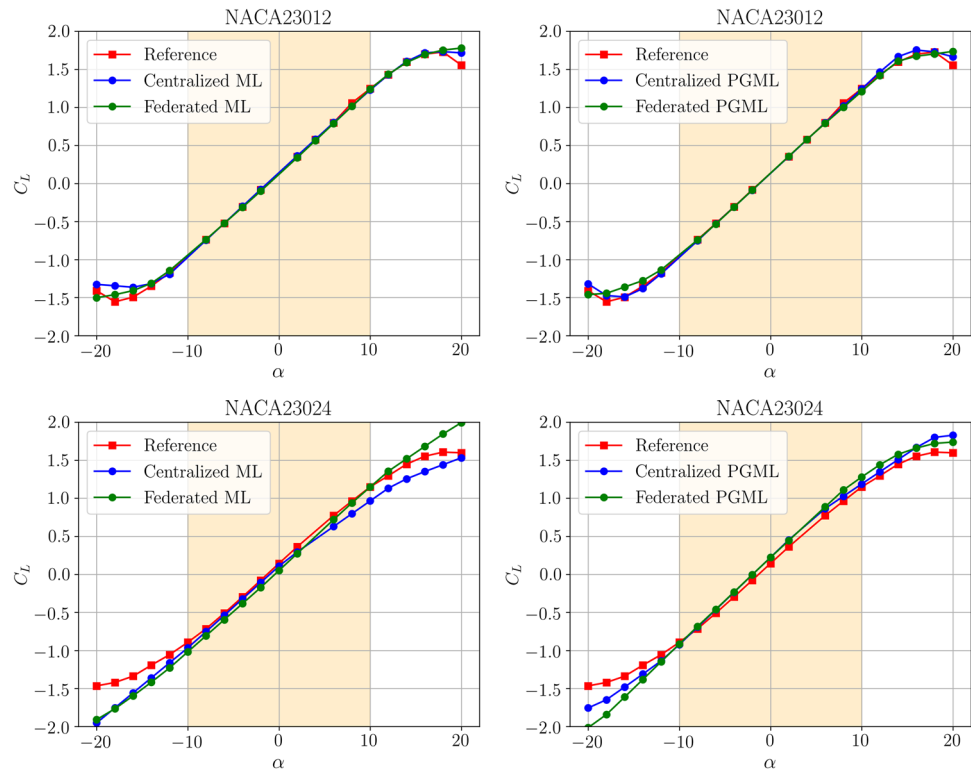


Figure 11. Predictive performance of the proposed centralized and federated models after 100 epochs.

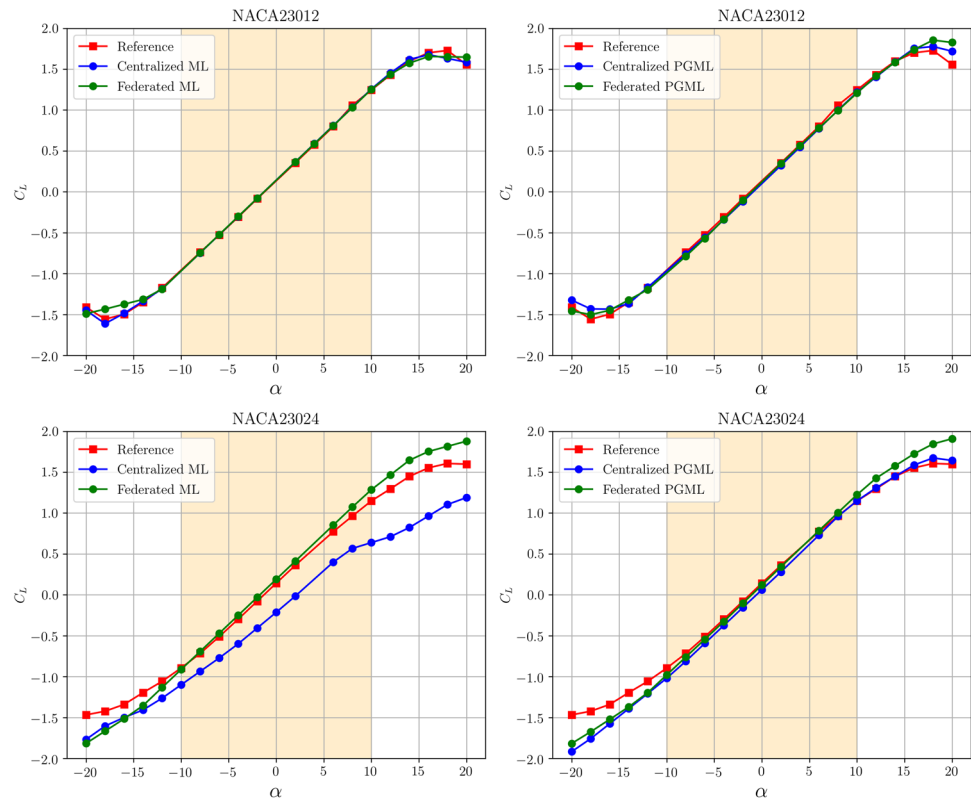


Figure 12. Predictive performance of the proposed centralized and federated models after 800 epochs.

training data. Our numerical experiments reveal that a federated model does not achieve equivalent accuracy compared to a model trained on centralized data collected from all clients. Instead, our investigation emphasizes the importance of refraining from making definitive claims regarding the equivalence of federated learning and centralized learning. This divergence in convergence behaviors is attributable to a combination of factors, including hyperparameter configurations, batch sizes, data distribution disparities, and the non-monotonic traits inherent to neural network-based approaches. Nevertheless, it is essential to highlight that the federated learning process enables the updating of a global model without disclosing locally collected data from diverse sources. In summary, we provided insight for the following challenges:

- Since the data generated by individual vendors never leaves the local servers, a decentralized setting guarantees better security, and addresses key concerns related to IPRs. Furthermore, since each vendor can focus on individual sub-component of the asset, the data quality can be significantly enhanced.
- Since each data owner can conduct the training using the limited amount of data that they generate and own, they can work with significantly low compute power in a cost-effective way.
- Since the data are converted into knowledge which is generally a highly compressed representation, the knowledge transfer can take place over very limited bandwidth.

Although the primary focus of this study is federated learning in the context of spatiotemporal reconstruction of dynamical systems, the research findings may enhance broader modeling and simulation software capabilities for designing capable and secure predictive digital twins for cross-domain simulations. We discuss the prospects of such decentralized digital twin (DDT) modeling paradigm in various computational science and engineering applications. The DDT approach is built on a federated learning concept to collaboratively learn an aggregated model while keeping all the training data on each client. Our results indicate that federated machine learning might be a key enabler for designing highly accurate DDT applications involving complex nonlinear spatiotemporal systems.

In summary, the trained models contributed by the different stakeholder can be reversed engineered thereby compromising data security. To create new surrogate models that are compatible with heterogeneous computer environments, in our future studies, we intend to use decentralized learning methodologies in the context of precision science and create novel distributed physics-guided federated learning approaches. Also, we would like to investigate which federated learning approach is best for combining weights. The demonstration cases handled here are relatively simple, but that was completely intentional as it eases the communication and dissemination of the work to a larger audience. However, the next logical step should be to apply the approach to a more complex problem.

Data availability

The synthetic data that supports the findings of this study is available within the article. The weekly sea surface temperature data can be found at the NOAA Optimum Interpolation (OI) SST V2 webpage: <https://psl.noaa.gov/data/gridded/data.noaa.oisst.v2.html>. The datasets used and/or analysed during the current study are also available from the corresponding author on reasonable request.

Received: 24 February 2023; Accepted: 8 November 2023

Published online: 16 November 2023

References

1. Westerman, G., Bonnet, D. & McAfee, A. The nine elements of digital transformation. *MIT Sloan Manag. Rev.* **55**, 1–6 (2014).
2. Arts, K., van der Wal, R. & Adams, W. M. Digital technology and the conservation of nature. *Ambio* **44**, 661–673 (2015).
3. Kapteyn, M. G., Pretorius, J. V. & Willcox, K. E. A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nat. Comput. Sci.* **1**, 337–347 (2021).
4. San, O., Rasheed, A. & Kvamsdal, T. Hybrid analysis and modeling, eclecticism, and multifidelity computing toward digital twin revolution. *GAMM-Mitteilungen* **44**, e202100007 (2021).
5. Rasheed, A., San, O. & Kvamsdal, T. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access* **8**, 21980–22012 (2020).
6. Bereska, L. & Gavves, E. Continual learning of dynamical systems with competitive federated reservoir computing. In *Conference on Lifelong Learning Agents*, 335–350 (organizationPMLR, 2022).
7. Bonawitz, K. *et al.* Towards federated learning at scale: System design. *Proc. Mach. Learn. Syst.* **1**, 374–388 (2019).
8. Ramu, S. P. *et al.* Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions. *Sustain. Cities Soc.* **79**, 103663 (2022).
9. Li, X., Jiang, M., Zhang, X., Kamp, M. & Dou, Q. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623* (2021).
10. Bacciu, D., Di Sarli, D., Faraji, P., Gallicchio, C. & Micheli, A. Federated reservoir computing neural networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–7 (IEEE, 2021).
11. Criado, M. F., Casado, F. E., Iglesias, R., Regueiro, C. V. & Barro, S. Non-IID data and continual learning processes in federated learning: A long road ahead. *Inform. Fus.* **88**, 263–280 (2022).
12. Kičić, I. *et al.* Adaptive learning of effective dynamics: Adaptive real-time, online modeling for complex systems. *arXiv preprint arXiv:2304.01732* (2023).
13. Chakraborty, S., Adhikari, S. & Ganguli, R. The role of surrogate models in the development of digital twins of dynamic systems. *Appl. Math. Model.* **90**, 662–681 (2021).
14. Sun, W., Lei, S., Wang, L., Liu, Z. & Zhang, Y. Adaptive federated learning and digital twin for industrial internet of things. *IEEE Trans. Industr. Inf.* **17**, 5605–5614 (2020).
15. Hartmann, D., Herz, M. & Wever, U. Model order reduction a key technology for digital twins. *Reduced-Order Modeling (ROM) for Simulation and Optimization: Powerful Algorithms as Key Enablers for Scientific Computing* 167–179 (2018).

16. Kong, L.-W., Weng, Y., Glaz, B., Haile, M. & Lai, Y.-C. Reservoir computing as digital twins for nonlinear dynamical systems. *Chaos: An Interdisciplinary J. Nonlinear Sci.* **33** (2023).
17. Chakraborty, S. & Adhikari, S. Machine learning based digital twin for dynamical systems with multiple time-scales. *Comput. Struct.* **243**, 106410 (2021).
18. San, O. The digital twin revolution. *Nat. Comput. Sci.* **1**, 307–308 (2021).
19. Research Dive, I. I. I. Digital Twin Market by Type (System Digital Twin, Product Digital Twin, and Process Digital Twin), Industry (Aerospace & Defense, Automotive & Transportation, Home & Commercial, Healthcare, Energy & Utilities, Oil & Gas, Agriculture, Telecommunication, and Others), and Regional Analysis (North America, Europe, Asia-Pacific, and LAMEA): Global Opportunity Analysis and Industry Forecast, 2021–2030. RA05951. howpublished <https://www.researchdive.com/5951/digital-twin-market> (2022). Last accessed 30 June 2022.
20. Gebbers, R. & Adamchuk, V. I. Precision agriculture and food security. *Science* **327**, 828–831 (2010).
21. Ashley, E. A. Towards precision medicine. *Nat. Rev. Genet.* **17**, 507–522 (2016).
22. Karpatne, A. *et al.* Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.* **29**, 2318–2331 (2017).
23. Childs, C. M. & Washburn, N. R. Embedding domain knowledge for machine learning of complex material systems. *MRS Commun.* **9**, 806–820 (2019).
24. Reichstein, M. *et al.* Deep learning and process understanding for data-driven Earth system science. *Nature* **566**, 195–204 (2019).
25. de Bezenac, E., Pajot, A. & Gallinari, P. Deep learning for physical processes: Incorporating prior scientific knowledge. *J. Stat. Mech: Theory Exp.* **2019**, 124009 (2019).
26. Willard, J., Jia, X., Xu, S., Steinbach, M. & Kumar, V. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 1–34 (2020).
27. Rai, R. & Sahu, C. K. Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (CPS) focus. *IEEE Access* **8**, 71050–71073 (2020).
28. Sansana, J. *et al.* Recent trends on hybrid modeling for industry 4.0. *Comput. Chem. Eng.* **151**, 107365 (2021).
29. Sharma, N. & Liu, Y. A hybrid science-guided machine learning approach for modeling chemical processes: A review. *AIChE J.* **68**, e17609 (2022).
30. Hoffer, J. G. *et al.* Theory-inspired machine learning-towards a synergy between knowledge and data. *Weld. World* **66**, 1291–1304 (2022).
31. Rajulapati, L., Chinta, S., Shyamala, B. & Rengaswamy, R. Integration of machine learning and first principles models. *AIChE J.* **68**, e17715 (2022).
32. Yang, Q., Liu, Y., Chen, T. & Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.* **10**, 1–19 (2019).
33. Zhang, C. *et al.* A survey on federated learning. *Knowl.-Based Syst.* **216**, 106775 (2021).
34. Wahab, O. A., Mourad, A., Otrok, H. & Taleb, T. Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems. *IEEE Commun. Surv. Tutorials* **23**, 1342–1397 (2021).
35. Liu, J. *et al.* From distributed machine learning to federated learning: A survey. *Knowl. Inform. Syst.* 1–33 (2022).
36. Ahmed, S. E., San, O., Kara, K., Younis, R. & Rasheed, A. Multifidelity computing for coupling full and reduced order models. *PLoS One* **16**, e0246092 (2021).
37. Chilson, P. B. *et al.* Moving towards a network of autonomous UAS atmospheric profiling stations for observations in the earth's lower atmosphere: The 3d mesonet concept. *Sensors* **19**, 2720 (2019).
38. Tran, D. T. *et al.* GANs enabled super-resolution reconstruction of wind field. In *Journal of Physics: Conference Series*, vol. 1669, 012029 (organization IOP Publishing, 2020).
39. Nolan, D. S., Zhang, J. A. & Stern, D. P. Evaluation of planetary boundary layer parameterizations in tropical cyclones by comparison of in situ observations and high-resolution simulations of hurricane isabel (2003). Part i: Initialization, maximum winds, and the outer-core boundary layer. *Monthly Weather Rev.* **137**, 3651–3674 (2009).
40. Mildrexler, D. J., Zhao, M. & Running, S. W. A global comparison between station air temperatures and MODIS land surface temperatures reveals the cooling role of forests. *J. Geophys. Res. Biogeosci.* **116** (2011).
41. Alavi, A. H. & Buttler, W. G. An overview of smartphone technology for citizen-centered, real-time and scalable civil infrastructure monitoring. *Futur. Gener. Comput. Syst.* **93**, 651–672 (2019).
42. O'Grady, M., Langton, D., Salinari, F., Daly, P. & O'Hare, G. Service design for climate-smart agriculture. *Inform. Process. Agric.* **8**, 328–340 (2021).
43. Chapman, L., Bell, C. & Bell, S. Can the crowdsourcing data paradigm take atmospheric science to a new level? a case study of the urban heat island of london quantified using netatmo weather stations. *Int. J. Climatol.* **37**, 3597–3605 (2017).
44. Sivaraman, V., Gharakheili, H. H., Fernandes, C., Clark, N. & Karlychuk, T. Smart IOT devices in the home: Security and privacy implications. *IEEE Technol. Soc. Mag.* **37**, 71–79 (2018).
45. Reda, H. T., Daely, P. T., Kharel, J. & Shin, S. Y. On the application of IoT: Meteorological information display system based on LORA wireless communication. *IETE Tech. Rev.* **35**, 256–265 (2018).
46. Ahire, D. B., Gond, D., Vitthal, J. & Ahire, N. L. IoT based real-time monitoring of meteorological data: A review. *Nitin L., IoT Based Real-Time Monitoring of Meteorological Data: A Review (February 25, 2022)* (2022).
47. Kudari, J. M. *et al.* Tools and techniques to implement AIoT in meteorological applications. In *Artificial Intelligence of Things for Weather Forecasting and Climatic Behavioral Analysis*, 53–73 (publisher IGI Global, 2022).
48. Ioannou, K., Karampatzakis, D., Amanatidis, P., Aggelopoulos, V. & Karmiris, I. Low-cost automatic weather stations in the internet of things. *Information* **12**, 146 (2021).
49. Zhang, J. *et al.* Towards building the federated GPT: Federated instruction tuning. *arXiv preprint arXiv:2305.05644* (2023).
50. Chen, Y., Qin, X., Wang, J., Yu, C. & Gao, W. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intell. Syst.* **35**, 83–93 (2020).
51. McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282 (organization PMLR, 2017).
52. San, O., Pawar, S. & Rasheed, A. Prospects of federated machine learning in fluid dynamics. *AIP Adv.* **12** (2022).
53. Li, T. *et al.* Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2**, 429–450 (2020).
54. Li, T., Sanjabi, M., Beirami, A. & Smith, V. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497* (2019).
55. Fallah, A., Mokhtari, A. & Ozdaglar, A. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948* (2020).
56. Deng, Y., Kamani, M. M. & Mahdavi, M. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461* (2020).
57. Tan, A. Z., Yu, H., Cui, L. & Yang, Q. Towards personalized federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* (2022).
58. Rowley, C. W. & Dawson, S. T. Model reduction for flow analysis and control. *Annu. Rev. Fluid Mech.* **49**, 387–417 (2017).
59. Reynolds, R. W. A real-time global sea surface temperature analysis. *J. Clim.* **1**, 75–87 (1988).
60. Reynolds, R. W. & Marsico, D. C. An improved real-time global sea surface temperature analysis. *J. Clim.* **6**, 114–119 (1993).
61. Kutz, J. N., Fu, X. & Brunton, S. L. Multiresolution dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.* **15**, 713–735 (2016).
62. Callahan, J. L., Maeda, K. & Brunton, S. L. Robust flow reconstruction from limited measurements via sparse representation. *Phys. Rev. Fluids* **4**, 103907 (2019).

63. Maulik, R., Egele, R., Lusch, B. & Balaprakash, P. Recurrent neural network architecture search for geophysical emulation. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–14 (organizationIEEE, 2020).
64. Robinson, H., Pawar, S., Rasheed, A. & San, O. Physics guided neural networks for modelling of non-linear dynamics. *Neural Netw.* **154**, 333–345 (2022).
65. Pawar, S., San, O., Aksoylu, B., Rasheed, A. & Kvamsdal, T. Physics guided machine learning using simplified theories. *Phys. Fluids* **33**, 011701 (2021).

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0019290.

Author contributions

O.S. and A.R. conceived the proposed research. O.S. and S.P. conducted the numerical experiments, analysed the data, and wrote the initial draft of the manuscript. All coauthors reviewed the manuscript, discussed the results, and contributed to the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to O.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023