Computational Mechanics (2023) 72:577–591
https://doi.org/10.1007/s00466-023-02279-x

ORIGINAL PAPER

A thermodynamics-informed active learning approach to perception and reasoning about fluids

Beatriz Moya[1] · Alberto Badías[2] · David González[1] · Francisco Chinesta[3,4] · Elías Cueto[1]

Received: 12 December 2022 / Accepted: 28 January 2023 / Published online: 23 February 2023

Abstract
Learning and reasoning about physical phenomena is still a challenge in robotics development, and computational sciences play a capital role in the search for accurate methods able to provide explanations for past events and rigorous forecasts of future situations. We propose a thermodynamics-informed active learning strategy for fluid perception and reasoning from observations. As a model problem, we take the sloshing phenomena of different fluids contained in a glass. Starting from full-field and high-resolution synthetic data for a particular fluid, we develop a method for the tracking (perception) and simulation (reasoning) of any previously unseen liquid whose free surface is observed with a commodity camera. This approach demonstrates the importance of physics and knowledge not only in data-driven (gray-box) modeling but also in real-physics adaptation in low-data regimes and partial observations of the dynamics. The presented method is extensible to other domains such as the development of cognitive digital twins able to learn from observation of phenomena for which they have not been trained explicitly.

Keywords Physics-informed incremental learning · GENERIC · Liquid simulation · Dynamic data driven application · Hybrid twins

1 Introduction

The research community has witnessed great advances in deep learning and artificial intelligence (AI) with respect to the imitation of human-like skills [33,43]. However, despite experiencing great growth, work is still in transition to the development of the so-called Artificial General Intelligence (AGI) [15]. In this next step of AI, we approach more anthropomorphic skills, independence, and high-level reasoning for decision making [48] and behavior learning [2,32]. One of the requirements to cross the boundary between AI and AGI is to improve the development of sensory and physics perception in machine systems. Physics perception refers to the interpretation and reasoning skills that interpret sensed data to make predictions. These technologies play an important role in robotics, especially in planning and controlling tasks [43]. To develop systems capable of reasoning about their surroundings, we need interactive real-time simulators of real-world physics, constructed by judicious mixing of theory, data, and computation [33].

In some sense, these systems resemble digital twins. However, these new systems, capable of perceiving and reasoning about physics, are more aligned with the concept of cognitive or hybrid digital twin [8]. Cognitive, or hybrid, twins are enriched with new measurements to correct the predictions that result from comparing the output of the simulation with the ground truth to optimize the accuracy of the solution in new scenarios that the user does not control. This

✉ Elías Cueto
ecueto@unizar.es

Beatriz Moya
beam@unizar.es

Alberto Badías
alberto.badias@upm.es

David González
gonzal@unizar.es

Francisco Chinesta
francisco.chinesta@ensam.eu

1 Aragon Institute of Engineering Research, Universidad de Zaragoza, Zaragoza, Spain

2 Polytechnic University of Madrid, Madrid, Spain

3 ESI Group Chair at the PIMM Lab, Arts et Mètiers Institute of Technology, Paris, France

4 CNRS@CREATE LTD, ENSAM Institute of Technology, 1 CREATE Way, Singapore 138602, Singapore

framework arises as a solution to complement current techniques in the so-called smart data paradigm, to profit from data and perform the desired update, adaptation, and knowledge enrichment promoting the efficient use of data.

This can be seen, alternatively, as a particular case of Dynamic Data Driven Applications Systems (DDDAS) [10,11]. In this context, data is introduced dynamically in a continuous learning loop to improve the performance of the obtained forecasting. Data-driven physics-informed simulators reach a higher generalization than unconstrained models or purely theoretical approximations. However, there are still difficulties in matching the model output to an evolving real physical environment [3]. For this reason, the learned simulator of real physics can be formulated as a hybrid twin, or more generally, a DDDAS, to adapt to new scenarios through continuous observation and measurement of labeled data.

Blakseth et al. propose the system *CoSTA* [6]. They develop a hybrid strategy to complement a physics-based model with a second data-driven term that will be in charge of learning the corrections. In [36], the authors proposed a hybrid twin of a hyperelastic beam with moving loads, displayed by means of augmented reality. In the field of perception and reasoning, Blakseth et al. [6] exploits DDDAs in scene understanding in unknown scenarios. Schenck and Fox [44,45] propose a system for physical reasoning about liquids that is corrected from observations without physical priors.

In this work, we take a sloshing fluid in a glass as the model problem. Despite its apparent simplicity, the fluid is highly deformable (something always difficult for perception systems), highly nonlinear, and dissipative (conservative problems have been demonstrated to be much easier to learn). We develop a method for the perception (tracking of the free surface of the fluid) and reasoning (providing the user with full-field information—velocity, stress, ...—predictions) about the physical state of a sloshing fluid instead of performing reconstruction and estimation of the liquid dynamics on based on images without physical knowledge [40].

Instead of developing a general integrator to perform physics simulation (see [1,5,42,47], for instance), our method constructs on the fly a learned simulator. This means that we learn the simulator to perform predictions based on the current state of the system. This learned simulator is initially built with synthetic full-field data from one single fluid. At runtime, when faced with video streams of a previously unseen surrounding reality, the simulator makes use of an active learning strategy in a thermodynamics-informed setting to correct systematic deviations of the observed reality from its predictions. Our approach ensures compliance to the first principles—conservation of energy, nonnegative entropy production—of the resulting simulations, even if they are constructed from partial observations of the reality (in our case, the observation of the free surface of the fluid).

Both the simulator originally learned and the active learning strategy are based on the use of the General Equation for Non-Equilibrium Reversible-Irreversible Coupling (GENERIC) formalism as an inductive bias to guide training according to the laws of thermodynamics [21]. Furthermore, the imposition of inductive biases could be advantageous in the correction process to reduce error bounds and adapt to new scenarios more efficiently [16]. The physical knowledge already gained in the source model is preserved for the sake of the physical interpretability of the results and the success of the adaptation. The proposed thermodynamically informed scheme will adapt to the inherent dissipative nature of real phenomena. Although the source algorithm is trained with nonmeasurable state variables required to capture the thermodynamical evolution of the system, the model can be corrected from the sole observation of the free surface of the new liquid if biased deviations are noticed.

As a result, we obtain a corrected, augmented intelligence system that performs the integration of the fluid dynamics in real-time from the evaluation of the free surface. The output is the prediction of the future states of the fluid as a result of its integration in time. This approach is coupled with a computer vision system to build a closed-loop algorithm that learns liquid slosh from real measurements. The adaptation is performed in an off-line phase to perform correction with online acquired data. After the correction, the simulation of the dynamics is accomplished in real-time, obtaining improved predictions of the phenomena taking place in the physical world.

The development of this method revolves around the convergence of the aforementioned DDDAS, active learning, and transfer learning strategies. Transfer learning profits from a model already trained to apply the gained knowledge to new tasks (in our case, previously unseen fluids). The goal is not to learn a new simulator without forgetting the basic features of fluid dynamics to compensate for the lack of measurements. This technique has also been considered in the a posteriori correction of models such as manifold learning in reduced order modeling of fluids [34] and dynamical optimization [19,22,30].

We propose a correction system that adapts to new, previously unseen, liquids to ensure an accurate simulation of their behavior. We train an initial model for glycerine. The algorithm, which is initially trained on full-field data, will be adapted to perform a physically sound integration with evaluations of the position of the free surface. Then, this system will make use of transfer learning to adapt to new liquids in scarce data regimes from the evaluations of the free surface of these new liquids without forgetting the thermodynamical insights already acquired in the first training. This work is structured as follows. Section 2 presents the formulation of the problem. Section 3 exposes the basic concepts of the thermodynamic formulation used in the algorithm. The source algorithm is presented in Section 4. Section 5 is dedicated to explaining the method in detail, from data acquisition to the correction algorithm. Section 6 shows the results of the method applied in real and computational settings. Finally, Sect. 7 summarizes the main results and conclusions observed in the application of the method.

2 Problem formulation

We describe the evolution of a complex physical system as a function of a set of state variables $\boldsymbol{s}_n$ that fully defines its thermodynamical state. Both the initial simulator and the correction algorithm use the GENERIC equation as an inductive bias [21]. GENERIC proposes a mesoscopic formulation of dynamical systems to describe their behavior in terms of the evolution of energy and entropy in the system. This approximation for dynamical systems can be discretized in time and its constituting terms can thus be inferred from the data [17]. The so-called Structure-Preserving Neural Networks (SPNNs) [24,25] are deep learning architectures that employ the GENERIC formalism as an inductive bias to learn dynamical patterns from data and perform simulations with physically meaningful guarantees.

Our method constructs in a first step a learned simulator trained offline with synthetic data from computational simulations [37]. This level of description is consistent with theories that have shown that human reasoning about fluid physics can be understood as a learned simulator operating at a coarse-grained level of description [4]. For this first simulator, we employ data from the simulation of full-field and high-fidelity smoothed particle hydrodynamics (SPH) simulations [35] of a source fluid—in our case, glycerine (viscosity $\mu = 0.950\,\text{Ns/m}^2$ and density $\rho = 1261\,\text{kg/m}^3$) [20]. We apply different initial velocities to the glass to trigger the slosh and build the training database. We employ the same glass and volume of liquid for all experiments because we are interested in adapting to different materials, but initial training and correction could be extended to new shapes and volumes.

Figure 1 describes the architecture of the original simulator. The final algorithm consists of three networks. First, we off-line train an autoencoder to find a low-dimensional manifold of the dynamics under study and run simulations under stringent real-time constraints [18].In this way, we compress the information of the initial dataset, which consists of the position, velocity, energy, and stress tensor of all the particles of the SPH discretization. This knowledge is transferred to two new networks. On the one hand, a network is dedicated to learning the physics of the problem. For this purpose, we train the time integrator of the evolution of the dynamics in the latent manifold based on structure-preserving NN. On the other hand, our learned simulator is faced with online measurements performed by a commodity camera. During this online phase, we only have access to the position of the free surface of the fluid. We substitute the encoder with a recurrent neural network to connect these partial measurements of the free surface of the liquid to the already unveiled latent manifold of the dynamics. The assembly of these three networks performs a simulation in the loop. The recurrent neural network is needed continuously to handle constant input from the camera.

However, this network is not accurate if we evaluate new fluids. In addition, it is unfeasible to learn a general simulator for any type of fluid that can be encountered in real life. We propose a correction algorithm that adapts to these new materials. The network will be corrected, but at the same time, it will preserve the knowledge gained in the previous training about fluid dynamics. This already acquired knowledge about thermodynamical insights of the fluid will be preserved to compensate for the limitations of the observations. Hence, our goal is not to enrich the model to be general to all fluids but to adapt to the new flow observed and profit from the proposed thermodynamical framework.

3 Physics-constrained learning

Once data are captured, either in computational, synthetic form or as a recording of the physical reality, our aim is to identify the state of the observed scene (perception) in terms of observable and non-observable variables—such as stresses, for instance—and to predict future states of the system (reasoning) by learned simulation. In the last years, a growing interest has been noticed in the incorporation of known physics in the form of inductive biases to this learning procedure. These representations will be essential for the generalization and interpretability of the method. Of particular importance, when some form of conservation (related to symmetries through Noether's theorem) is present, this can be imposed on the learning procedure by invoking Lagrangian or Hamiltonian frameworks. Despite their success, many dynamical systems are also compromised by dissipative effects, which implies that this framework of techniques is no longer valid for application. This premise also matches the dissipative nature of the real world and the entropy that arises from the lack of information. Fluid dynamics, as well as other dissipative dynamical systems, has a so-called metriplectic structure [28]. This formalism is suitable for cases in which the conservative Hamiltonian description of a dynamical system includes unresolved degrees of freedom that are not included in mesoscopic descriptions and that introduce dissipation by the fluctuation-

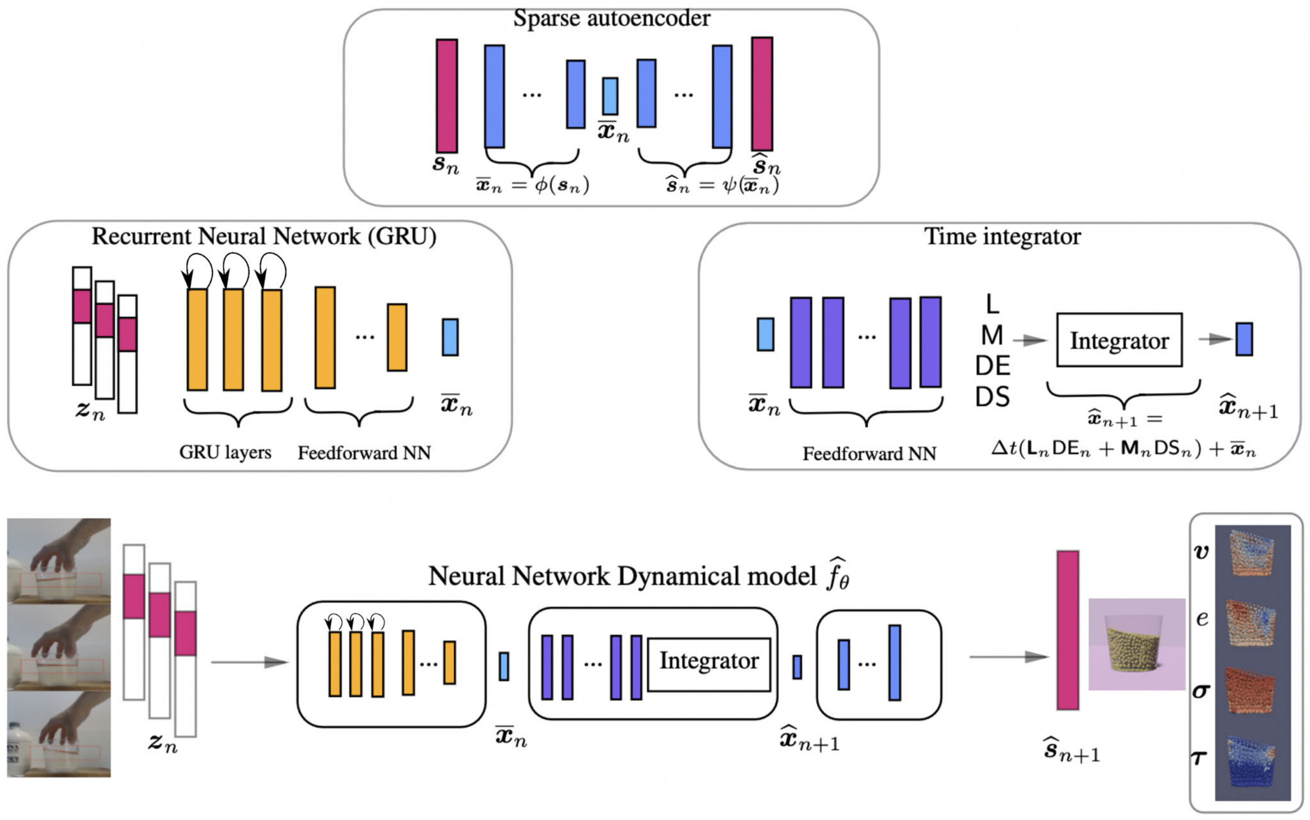

Fig. 1 Initial learned simulator. It is composed of three parts. We first train an off-line autoencoder from synthetic full-field data s_n to find a low-dimensional manifold of the dynamics under study. (Left) There we train a structure-preserving network to determine its constituents: the symplectic matrix $\boldsymbol{L}$, the dissipation matrix $\boldsymbol{M}$, an approximation to the gradient of energy, DE, and an approximation to the gradient of entropy, DS. (Right) Since we only have access to the free surface of the fluid (pink entries z_n in the input vectors s_n), a recurrent neural network exploits knowledge in sequences of data to find the embedding to the low-dimensional manifold discovered off-line from full-field data (transfer learning). The Structure-Preserving Neural Network (SPNN) propagates in time the evolution of the dynamics in the manifold, fulfilling the principles of thermodynamics as a major requirement. Finally, the autoencoder decoder projects the new state of the fluid in the world coordinates to recover the volume of the liquid and other quantities of interest ($\boldsymbol{v}$ velocity, e energy, $\boldsymbol{\sigma}$ normal stress, $\boldsymbol{\tau}$ shear stress, grouped in the full field vector s_{n+1}), whose knowledge is used to integrate in the latent manifold

dissipation theorem [29]. Consider an initial microscale description at the molecular dynamics scale that can be expressed in terms of a purely conservative formulation. The degrees of freedom, and therefore knowledge, that we omit growing from the micro to the meso and macro scales introduce dissipation that is included in the metric part of the formulation.

This structure-preserving formulation guarantees the conservation of critical quantities (mass and momentum) and the thermodynamical admissibility of the evolution of the system under study. GENERIC is the thermodynamic framework chosen to derive a structure-preserving time integrator to describe the time evolution of a system regarding the evolution of its energy and entropy functionals [21]. As a consequence, this framework is also valid for dynamical systems that go beyond equilibrium in a thermodynamical context. The GENERIC formulation is usually written as follows:

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = \boldsymbol{L}\nabla E + \boldsymbol{M}\nabla S.$$

Here $\boldsymbol{s}$ denotes a set of independent state variables that fully describe the thermodynamical state of the fluid. Without that information, we lack a GENERIC structure. Fluid dynamics is fully described in terms of the position and momentum of the liquid particles, the internal energy, and, in the case of learning more complex fluids, the extra-stress tensor related to their microscopic evolution [14]. $E(\boldsymbol{s})$ and $S(\boldsymbol{s})$ are the global energy and entropy of the system. $\boldsymbol{L}(\boldsymbol{s})$ is the Poisson matrix. It is skew-symmetric and, together with the energy gradient ∇E, characterizes the reversible part of the dynamics studied. $\boldsymbol{M}(\boldsymbol{s})$ is the friction matrix, which describes the dissipative irreversible characteristics of the system in conjunction with the entropy gradient ∇S. $\boldsymbol{M}$ is symmetric positive semidefinite.

On top of the formulation and matrix descriptions mentioned above, we must also ensure the fulfillment of the so-called *degeneracy conditions*:

$$\boldsymbol{L}\frac{\partial S}{\partial \boldsymbol{s}} = \boldsymbol{M}\frac{\partial E}{\partial \boldsymbol{s}} = \boldsymbol{0},$$

that guarantee that the energy E is not involved in the production of entropy of the dissipative part of the dynamics, and that entropy S does not contribute to the conservation of energy.

The GENERIC equation can be discretized in time by a forward Euler scheme formulated in time increments Δt, and the resulting formulation is subsequently inferred from the data:

$$\boldsymbol{s}_{n+1} = \boldsymbol{s}_n + \Delta t \left(\mathsf{L}_n \frac{\mathsf{D}E_n}{\mathsf{D}\boldsymbol{s}_n} + \mathsf{M}_n \frac{\mathsf{D}S_n}{\mathsf{D}\boldsymbol{s}_n} \right),$$

subjected to the degeneracy conditions:

$$\mathsf{L}_n \frac{\mathsf{D}S_n}{\mathsf{D}\boldsymbol{s}_n} = \boldsymbol{0}, \quad \mathsf{M}_n \frac{\mathsf{D}E_n}{\mathsf{D}\boldsymbol{s}_n} = \boldsymbol{0}, \tag{1}$$

ensuring the thermodynamical consistency of the resulting model.

An explicit time integration algorithm could be jeopardized by a selection of an inappropriate time step. In our work, the maximum time step is constrained by the data acquisition and real-time representation frequency of the camera. The camera interacts with reality at 60 Hz, which corresponds to a time step $\Delta t = 0.017$ seconds. This time step is sufficiently small to capture the evolution of the slosh and ensure the stability of the time integration. In addition, synthetic data was obtained using a stable time step. Hence, the neural network is trained over a stable dynamical evolution. In any case, Romero and coworkers have deeply studied the stability of time integration schemes based on the GENERIC formalism. Their conclusion is that GENERIC-based integrators are robust and stable even for large time steps, see [39].

Structure-preserving neural networks (SPNN) embed GENERIC in a deep learning architecture to reveal the value of the main elements of equation [24,25,31,49]. In this framework, discretized gradients of energy and entropy are targets of network optimization. Additionally, although **L** and **M** are generally known in the literature, they are unknown in the low-dimensional manifold where we simulate the dynamics. Therefore, the network learns **L**, **M**, $\frac{\mathsf{D}E_n}{\mathsf{D}\boldsymbol{s}_n}$, $\frac{\mathsf{D}S_n}{\mathsf{D}\boldsymbol{s}_n}$ from the data. This is represented in Fig. 1.

4 Initial learned simulator

The initial glycerine simulator is trained with synthetic data coming from the simulation of a fluid discretized into particles based on the smoothed particle hydrodynamics technique. This coarse level of description is appropriate to have sufficient information about the dynamics to train learned simulators without compromising time. The state of the fluid is described by a set of state variables (position, velocity, energy, and stress tensors of each particle) evaluated in each particle of the discretization.

Due to the complexity of the problem, the simulator involves three different steps, and transfer learning is applied to carry the knowledge from simpler architectures to more elaborate ones. First, we train a fully connected autoencoder to embed the data in a low-dimensional manifold where we will perform the integration in time. It consists of two parts: the encoder and the decoder. The encoder learns a mapping $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^d$ to a low-dimensional manifold where the dynamics are embedded. The decoder $\psi : \mathbb{R}^d \rightarrow \mathbb{R}^D$ has the same structure as the encoder, but is inverted, to project the state vectors of the latent space to the full order space. Not only will the computational cost for training the integrator be reduced, but the pre-processing will also facilitate the learning process. By applying model order reduction, we are already triggering the emergence of the patterns and encouraging the network to learn the main features of data. In spite of the correlations between the state variables, one single fully connected autoencoder resulted inefficient to learn the features of all the state variables given the size of the problem. Instead, we train five autoencoders in parallel, one per each group of the state variables already mentioned, to capture the intrinsic insights of each of them.

The final latent manifold results from the merging of the latent subspaces of each autoencoder. The bottleneck of each autoencoder is truncated to reduce the dimensionality by forcing sparsity on its learning. Thus, for a given bottleneck, only a few latent values have a similar order of magnitude greater than zero. The decoder mirrors the encoder. The resulting latent manifold has 13 dimensions. The decoder is part of the new model, but it is not considered in the correction. Sparsity is imposed with a L_1 regularizer that penalizes activation in the bottleneck to enforce a higher reduction. This penalization complements the mean squared error reconstruction loss of the solution of the autoencoder:

$$\mathcal{L}_{AE} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{s}_i - \hat{\boldsymbol{s}}_i)^2 + \lambda_{\text{reg}}\sum_{i=1}^{N}|\boldsymbol{x}_i|.$$

Sparsity loss is weighted by a lambda factor to control its influence on the learning process. In the source model, the weight factor is defined within the range $\lambda_{\text{reg}} = 0.001 - 0.005$. As we train an autoencoder for each group of state variables, it is chosen according to each case.

The full-field state of the fluid cannot be evaluated. That is, we cannot measure with a camera the internal energy of points of the fluid; we train a mapping from the available information of the free surface to the latent manifold in the second step of the procedure. Therefore, we transfer the information from the previously obtained latent manifold to a recurrent neural network to substitute the encoder. The recurrent neural network exploits the analysis of sequences of data to distill insights about the dynamics that enable one to find a mapping between the observations and the latent manifold, which contains complete information about the embedded dynamics. We employ a gated recurrent unit (GRU) structure [9]. This architecture will be trained with the decoder frozen. We need a sequence of at least 16 snapshots to correlate the measurements of the free surface with the latent manifold. The net consists of three GRU hidden layers of 26 neurons, and one last fully connected feedforward layer to connect the last GRU layer to the latent space of size 13. The sequence used contains the snapshot at time t, and the 15 previous snapshots. The network learns a mapping from each snapshot sequence to the representation of the state of the snapshot at time t in the latent manifold. We evaluate the precision of the mapping by measuring the discrepancy between the output of the GRU, the predicted reduced order representation, and the ground truth latent vector:

$$\mathcal{L}_{\text{GRU}} = \frac{1}{N_{\text{snap}}} \sum_{n=1}^{N_{\text{snap}}} (\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n)^2.$$

The architecture designed for the GRU was also tested with vanilla RNN and LSTM networks. Although ordinary RNN did not succeed in finding a mapping, LSTM reached a performance similar to that of GRU. However, we decided to use the GRU for simplicity.

Finally, the structure-preserving neural network is trained for time integration of the dynamics based on the reconstruction accuracy of the solution in the latent manifold, as well as the compliance of the degeneracy conditions. This network is composed of 13 hidden layers of size 195 with ReLU activations but for the first and last layers, which have linear activations. The loss $\mathcal{L}_{\text{SPNN}}$ used for the training includes the mean squared error reconstruction error of the prediction $\mathcal{L}^{\text{mse}}_{\text{SPNN}}$ and the loss related to the degeneracy conditions $\mathcal{L}^{\text{deg}}_{\text{SPNN}}$. The reconstruction error is weighted by a λ factor to prioritize this term. In this model, $\lambda_{\text{SPNN}} = 10^3$. In other words,

$$\mathcal{L}^{\text{mse}}_{\text{SPNN}} = \frac{1}{N_{\text{snap}}} \sum_{n=1}^{N_{\text{snap}}} (x_{n+1} - \hat{x}_{n+1})^2,$$

$$\mathcal{L}^{\text{deg}}_{\text{SPNN}} = \frac{1}{N_{\text{snap}}} \sum_{i=1}^{N_{\text{snap}}} (\mathsf{L}_n \mathsf{DS}_n)^2 + (\mathsf{M}_n \mathsf{DE}_n)^2,$$

$$\mathcal{L}_{\text{SPNN}} = \lambda^{\text{mse}}_{\text{SPNN}} \mathcal{L}^{\text{mse}}_{\text{SPNN}} + \mathcal{L}^{\text{SPNN}}_{\text{deg}}.$$

The parameters used for reproducibility are shown in Table 1.

5 Method

5.1 Data acquisition

5.1.1 Computational datasets

First, we perform a set of purely computational experiments. This allows us to obtain accurate error measurements against high-fidelity simulations that are considered ground truth. Synthetic data are obtained from computational simulations using the software Abaqus (Dassault Systèmes, Simulia Corp).

The method was tested against a total of ten different liquids presenting both Newtonian and non-Newtonian behavior. We initially train a learned glycerin simulator [20]. With this simulator, we try to understand the behavior of water and melted butter, for instance, both described as Newtonian fluids. Although blood is sometimes considered Newtonian, it has been described as non-Newtonian [46]. It is considered to be shear thinning, becoming less viscous under the stress applied. The general description of the rheology and change of fluid properties is made based on the Herschel–Bulkley model [26]

$$\boldsymbol{\tau}(t) = k\boldsymbol{\gamma}n(t) + \boldsymbol{\tau}_0,$$

Table 1 Training parameters of the source model

	lr	*wd*	Hidden layers	Input size	Output size	λ	Epochs
Autoencoder q	1e−4	1e−6	$N = 2$ of 120 neurons	6402	20	2000	10,000
Autoencoder v	1e−4	1e−5	$N = 4$ of 200 neurons	6402	20	2000	10,000
Autoencoder e	1e−4	1e−5	$N = 3$ of 40 neurons	2134	10	2000	10,000
Autoencoder σ	1e−4	1e−5	$N = 3$ of 200 neurons	2134	20	2000	10,000
Autoencoder τ	1e−3	1e−6	$N = 3$ of 200 neurons	6402	20	2000	10,000
SPNN	1e−3	1e−5	$N = 13$ of 195 neurons	13	195	1000	5000
GRU	1e−3	1e−5	3 GRU × 26 + 1 FC × 13	16 × 42	13	–	10,000

where $\boldsymbol{\tau}$ is the shear stress, $\boldsymbol{\tau}_0$ the yield stress, $\boldsymbol{\gamma}$ the shear rate, k the consistency index, and n the flow index. For shear-thinning fluids, $\boldsymbol{\tau}_0$ is greater than 0, and the flow index is $n > 1$. In the proposed case of computational blood, the computational liquid is defined by the constants $k = 0.017 Pas$, $n = 0.708$, and $\boldsymbol{\tau}_0 = 0$.

These data sets are publicly available at https://github.com/beatrizmoya/sloshingfluids/.

5.1.2 Real world dataset

We then employed a stereo camera for the acquisition of data of the free surface of real-world liquids. Although there are known applications for sophisticated tools such as PIV cameras [7,38], we assume that one only has access to the position of the free surface with an ordinary stereo camera. The camera chosen is a Real-Sense D435 https://www.intelrealsense.com/depth-camera-d435/. We deliberately omit the possibility of using PIV systems, for instance, to force our system to work in a partial data regime, i.e. without any velocity data other than the reconstruction of the free surface.

In a camera model, the correlation of pixels in a picture, or frame, with the point they represent in the real world $\boldsymbol{p}_w$ is given by extrinsic and intrinsic parameters. Extrinsic parameters, rotation $\boldsymbol{R}$, and translation $\boldsymbol{t}$ matrices define the position and orientation of the camera referring to the world frame. It outputs the relative position of the camera in the real world. On the other hand, the intrinsic parameters $\boldsymbol{K}$ map the pixel coordinates to the camera coordinates. The camera provides the intrinsic and extrinsic parameters to convert the pixel coordinates on the image directly to three-dimensional coordinates they represent in the real world,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

or, in brief,

$$\tilde{\boldsymbol{x}}_s = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\boldsymbol{p}_w.$$

The camera has stereoscopic depth technology to perceive the depth field of every pixel in the frame. With two lenses, triangulation [23] gives the 3D position of the points.

Despite the accuracy of the camera, fine-tuning was necessary due to the lack of texture of the glass and liquids. These conditions difficult the tracking points, resulting in errors. The application of filters (hole-filling and edge-preserving filters) improves the performance of depth estimation. In optimal conditions, the error in depth reconstruction is lower than 2% in a 0–2 m range. After the fine-tuning of the camera, in the threshold where we locate the glass for the recording, the camera noise in depth estimation is of the order of two millimeters.

Given a sufficient quality of depth resolution, we perform the observation and tracking of the free surface. We extract the profile of the free surface from binary (black and white) images. Under appropriate adjustment, this representation shows a black-to-white gradient between the liquid and the environment, see Fig. 2. These color changes enable tracking the free surface. We extract the depth maps of the pixels of the aforementioned boundary. No further smoothing of the data is performed. Real-time tracking is performed with a

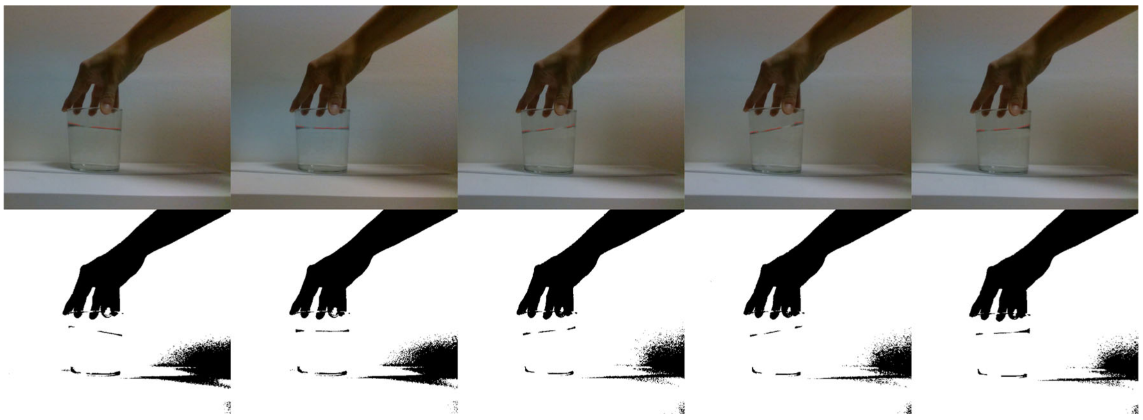

Fig. 2 Representation of surface tracking. Each color frame is converted to a binary frame in which the surface is detected (shown in red in the corresponding color frame). The depth map is built upon these points. For transparent liquids, such as the glass of water shown in the picture, the depth map is of lower quality, resulting in incomplete detection of the surface

camera resolution of 480 × 640 pixels.

From the acquired map, we project the pixel coordinates of the free surface to real-world coordinates and select their horizontal and vertical coordinates, or 2D real coordinates. Given the density of points detected on the free surface, we perform a simple linear interpolation to output the vertical displacement at equally spaced points of the free surface. This state information of the free surface is assembled in sequences to feed the source algorithm.

The data set of the free surface measurements performed on real liquids has been made publicly available: https://github.com/beatrizmoya/RLfluidperception

5.2 Correction algorithm

As already stated, our system is initially trained to perceive and reason about glycerine. However, it then faces different, previously unseen fluids. To provide useful and credible predictions for these new fluids, an active learning strategy is employed.

The sloshing dynamics is described by its dynamical state $\boldsymbol{s}_n \in \mathcal{S}$, where $\mathcal{S}$ represents the state space in which the dynamics are embedded. From that state $\boldsymbol{s}_n$, we have access to a partial observation $z_n \subset \boldsymbol{s}_n$. If $\boldsymbol{s}_n$ refers to the fully dynamical description of the slosh, z_n represents the degrees of freedom observed on the free surface. Free surface detection and tracking are performed employing computer vision techniques. The slosh evolves over time, and the integrator provides results for future states $\boldsymbol{s}_{t+1}$, not only for the observed states z_{t+1}. For its calculation, a fully connected neural network based on the GENERIC formalism estimates the parameters of the equation. The ultimate objective of this work is to correct the existing model so that we obtain a satisfactory estimate of the GENERIC parameters, which results in an accurate reconstruction of the motion of the liquid.

The loss used for adaptation is related to the accuracy of the free surface motion reconstruction. We choose an L_2 norm for this purpose. We also seek solutions that fulfill the degeneracy conditions established in the GENERIC formalism. As a result, the loss is composed of these two terms, adequately weighted to control the influence of each term on the optimization of the network. The degeneracy condition is thus considered a soft constraint:

$$\mathcal{L}_{\text{correction}} = \lambda \frac{1}{N} \sum_{N} \left\| z_{n+1} - \hat{z}_{n+1} \right\|^2 + \frac{1}{N} \sum_{N} \left(\left\| \mathsf{L}_n \frac{\mathsf{D}E_n}{\mathsf{D}s_n} \right\|^2 + \left\| \mathsf{M}_n \frac{\mathsf{D}S_n}{\mathsf{D}s_n} \right\|^2 \right). \quad (2)$$

In each correction loop, the algorithm is fed with N data samples. With them, the network parameters are updated. This process is repeated until we reach convergence in the adaptation process.

Algorithm 1 Update pseudocode

```
Require: Free surface information $z_n \in Z$ as a subset of the full state $s_n \in S$, and the source model $\pi_\theta$
Ensure: Next dynamical states in time $s_{n+1}$
  for Iterations until convergence do
    for n=1 to N sequences do
      Encoder $\overline{x}_n \leftarrow \phi_{\text{GRU}}(z_n)$;
      Compute forward propagation and determine action $a_n \leftarrow \pi_\theta(\overline{x}_n)$, with $a_n = [\mathsf{L}_n, \mathsf{M}_n, \mathsf{DE}_n, \mathsf{DS}_n]$;
      Determine next integration step $\widehat{x}_{n+1} \leftarrow \Delta t(\mathsf{L}_n \mathsf{DE}_n + \mathsf{M}_n \mathsf{DS}_n) + \overline{x}_n$;
      Decoder $\widehat{s}_{n+1} \leftarrow \psi(\widehat{x}_{n+1})$;
    end for
    Extract free surface of $\widehat{s}_{n+1}$;
    Compute loss $\mathcal{L}_{\text{correction}}$;
    Backward propagation and update behavior of $\pi_\theta$;
  end for
  return Optimized perception and reasoning approximation $\pi_\theta$
```

As described in the algorithm, we first collect information from the observation of a fluid. Then, we compute the actions to obtain L, M, $\frac{\mathsf{D}S_n}{\mathsf{D}s_n}$, $\frac{\mathsf{D}E_n}{\mathsf{D}s_n}$ and perform the time integration for each given state. We then computed the loss with this information and perform backpropagation to update the neural network.

The backpropagation is computed through selected layers of the whole source model. This approach, in conjunction with low learning rates, ensures a minimum loss of information from the previous model to preserve the insights already learned with GENERIC. We profit from these known patterns to deal with the limitations coming from partial observations and low-data regimes.

6 Results

6.1 Cognitive digital twins in a purely computational scenario

The employ of pseudo-experimental data coming from simulations allows us to compute precise error measurements on the different adaptation strategies and their effect on diverse quantities of interest. Given a learned simulator trained offline for glycerine data, we first apply the updated methodology to learn three different liquids presenting rather diverse behaviors: water and butter as Newtonian fluids, and blood, simulated as a non-Newtonian liquid. We employ synthetic data consisting of snapshots taken from four simulations for each liquid performed under different velocity conditions to trigger various sloshing results. From this information, we prepare the dataset with sequences of the positions of the particles that belong to the free surface. That is, we track the particles on the free surface, take their positions, and prepare sequences of 16 snapshots for each time step Δt (the time instant of interest and the 15 previous states of the free surface). The length of 16 was found to be the minimum length of the sequences to find an embedding of the free surface measurements on each fluid's latent manifold, already computed offline. The water data set has 750 snapshots in total and butter and blood have 480. Synthetic data are available at a sampling frequency of 200 Hz, which is a time step of $\Delta t = 0.005$ s. Therefore, the source model was built for this availability of data and time step. However, the snapshots are sampled at every frequency of 60 Hz, or equivalently $\Delta t = 0.015$ s, which matches the performance frequency of the commodity camera that will be used in the real scenario. Therefore, the change in the time step also affects the model that should be corrected. The sequences of the data set are randomly split into two subsets: 80% for training and 20% for testing.

We carry out the correction in the integration scheme and the embedding onto a lower-dimensional manifold. The correction is accomplished by activating the backpropagation in the last layer of the GRU network, and the four last layers, out of 13, of the SPNN. Since these are precedent layers of the decoder, and we train the network as a whole, we achieve the desired reconstruction without the need of altering the last structure of the network. The adaptation converges after 2000 epochs, at a small learning rate $lr = 0.0005$ and weight decay $wd = 0.00001$. We choose Adam optimizer [27]. The reconstruction is weighted by a factor $\lambda = 2000$. Figure 3 shows the transition from the original manifold, trained offline, to the new latent space that fits the emulated dynamics. In particular, we show the first three state variables of the latent vector z, which correspond to the latent variables of position. Because the liquids bear similarity to the original liquid, glycerine, their manifolds show no drastic changes in their structure compared to those of the initial solution. This fact also highlights the generality of the dynamics patterns learned in the source glycerine simulator.

The height reference is placed at the bottom of the glass and, therefore, the relative error is calculated with the total height of the points on the free surface selected for the evaluation:

$$\text{error} = \frac{1}{N}\sqrt{\sum_{n=1}^{N}\frac{z_n^2 - \hat{z}_n^2}{z_n^2}},$$

with N the number of samples in the data set and z_n and $\hat{z}_n$ the ground truth and the simulated free surface data, respectively.

The relative error is represented in Fig. 4. The update algorithm improves the performance of the simulator, significantly reducing the reconstruction error of the free surface. The relative error is also presented for the complete data set

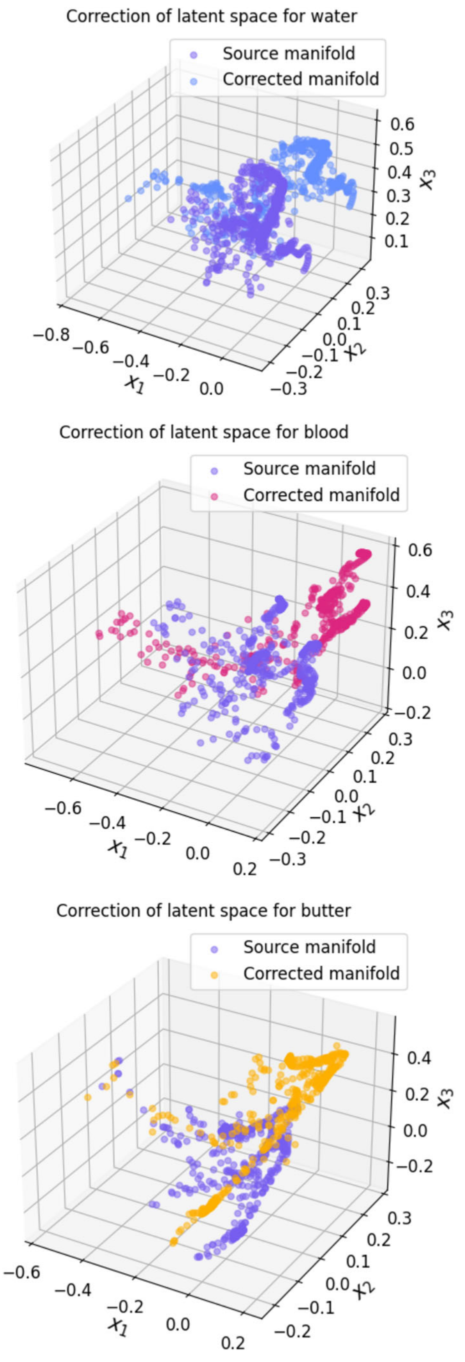

Fig. 3 Representation of the correction of the latent manifold of the position. The latent representation evolves to match the features of the new liquid

of each liquid, reconstructing the 80% used in the correction and the 20% considered for testing purposes. The four simulations can be distinguished in the graph. The error peaks appear in the top peaks of the sloshing of each simulation, where the surface deformation is greater. However, the error continues to be in the low range.

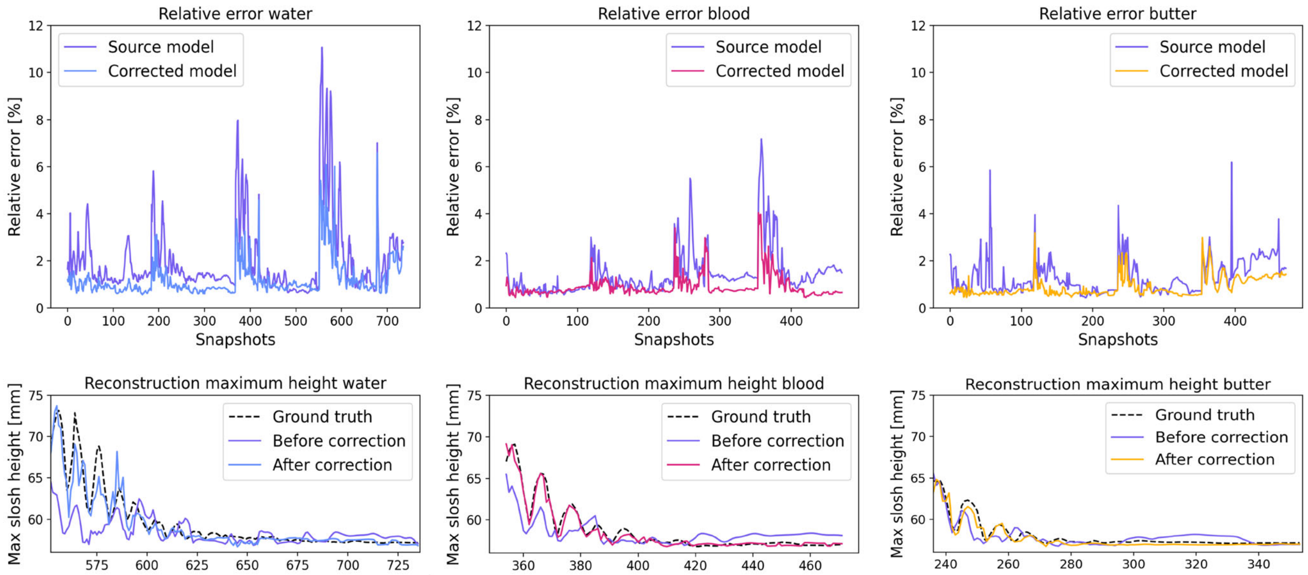

Fig. 4 (Top) Relative error of the reconstruction of the free surface for the four simulations used for training and test. The relative error obtained with the source model is reduced after optimization. (Bottom) Detail of the sloshing height reconstruction of the most critical simulation. The method correctly emulates the behavior not only in magnitude terms but also, notably, in the precise time occurrence of the peaks

Water presents a higher sloshing than glycerine. Thus, the desired solutions were out of the source database. In this case, the maximum error drops from 12 to 4–6%, remaining at an average of not more than 3%. Butter reconstruction error is reduced from 6% to less than 2%. Finally, the algorithm manages accurately to evolve from a Newtonian to a non-Newtonian fluid to replicate the behavior of blood. For this liquid, the maximum relative error goes from 7 to 3.5%. It is worth mentioning that, provided that we employ a meshless method (SPH) to generate synthetic data, we can observe some errors coming from the different particle distributions. Although the movement matches the ground truth, we can have two different particle configurations. This results in different interpolated free surfaces that cause small errors despite their strong resemblance.

We analyze the performance of the correction by evaluating the maximum height of the slosh in the most critical simulation for each liquid in Fig. 5 This detail is correlated with the correspondent segment of the relative error graphs. Finally, it is worth highlighting that the reconstruction adjusts better not only to the magnitude of the slosh but also to their occurrence in time. Before the correction, some peaks appear to be delayed or skipped.

6.2 Adaptation to a real-world scenario

Employing pseudo-experimental data coming from simulations has the great advantage of allowing the computation of precise error measurements. However, the ultimate objective of this work is to develop a methodology for real-life scenarios. In this scenario, the algorithm must adapt to previously unseen real liquids with different properties that result in different frequencies and magnitudes of slosh and dissipation and duration of slosh. This correction will be done by evaluation of the free surface of liquids tracked by using computer vision. This problem presents various challenges, such as errors in depth estimation, difficulty in detecting and tracking the free surface, the complexity of real liquids to capture all the dynamical features of the recorded videos, and user actuation. The last statement refers to the direction of movement of the liquid. In this work, we limit the experiment to a plane movement in two possible directions (move the glass to the left or right), but we can experience slight deviations in the actuation. We impose this restriction to bound the problem to a case where we can properly evaluate the free surface from a fixed position of the camera with the detection method employed.

Four different liquids are evaluated in the updated framework proposed here: water, honey, beer, and gazpacho (traditional Spanish cold soup). These four materials cover a wide spectrum of behaviors and textures, from lower to higher viscosities compared to glycerine. In addition, the liquids selected are daily products of interest in real manipulation that can show different behaviors depending on the production process. The glass used is of the same shape as the computational container used in the simulations of the source model, and the vessel is filled approximately to the same height.

There are two recordings for each selected liquid. The first video, divided into 80% snapshots for training and 20% for

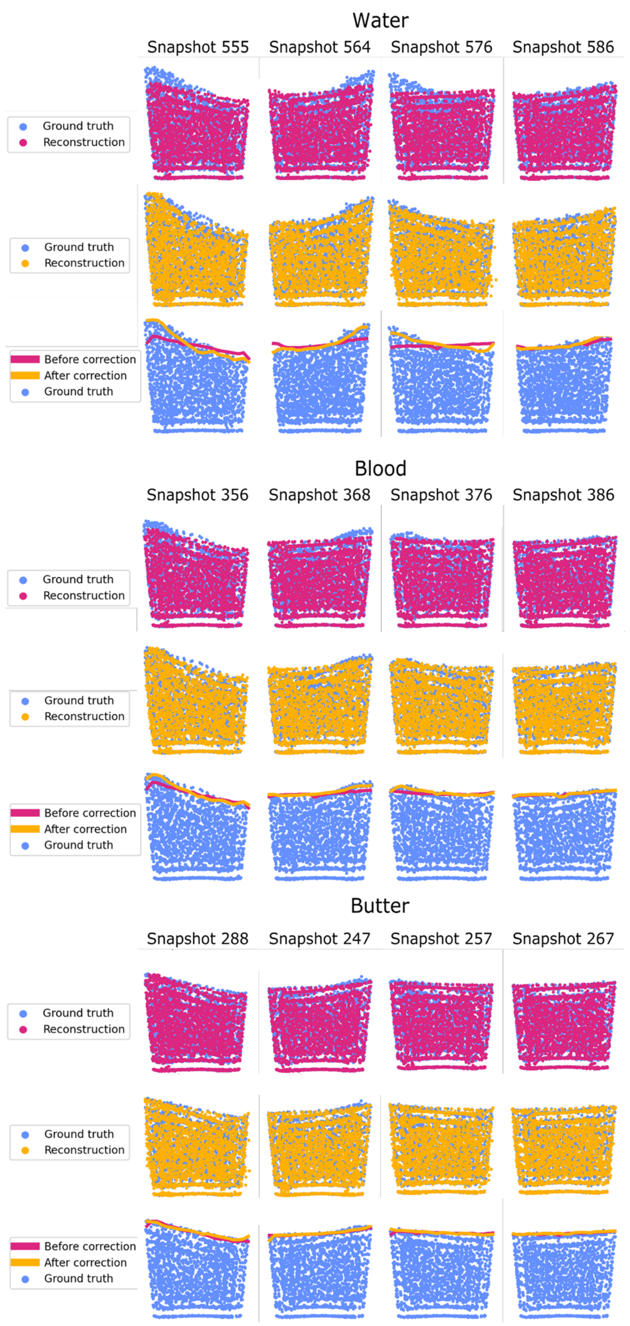

Fig. 5 Fluid reconstruction before and after correction of water, blood, and butter. Representation of the sloshing peaks of ground truth (in blue) compared to the network results before (in red) and after (in yellow) applying the correction to adapt to observations in selected critical snapshots, indicated in the figure. (Color figure online)

validation to prevent overfitting, is used in the update algorithm. The second video evaluates the performance of the new models on unknown datasets. Given these data, we apply the correction scheme for each liquid individually, obtaining four new simulators, adapted for each scenario. Similarly to the computational test case, correction is performed in selected active layers. Considering the complexity of the liquids' behavior and the noise of the samples, the last layer of the recurrent neural network (GRU) and the 5 last layers, out of 13, of the SPNN are activated for the correction. Correction of the recurrent neural network will adapt the manifold to the new liquid, while partial activation of the SPNN will adapt the simulation to the new dynamics observed. The real benchmarks require more correction than the computational benchmarks. However, we maintain low learning rates and partial activation to prevent the network from learning the measurements' noise and forgetting the previously learned dynamical patterns. The correction converges after 4000 epochs, with a small learning rate $lr = 0.0001$ and weight decay $wd = 0.00001$. The reconstruction is weighted by a factor $\lambda = 2000$.

Figure 6 shows the results of the correction in the training and testing recordings of the four liquids. The four exhibit an improvement in the sloshing reconstruction compared to the performance of the source model before correction. Furthermore, the temporal integration with data from the test datasets presents a notable performance considering that this information is new to the network. The training recordings are shorter than 10 seconds and only three or four sloshes are captured in these datasets. However, the new model learns the new target behavior. Although we work in a low-data regime to perform complex liquid correction, the method successfully reproduces the train and test benchmarks. These results are obtained because of the inductive biases learned and preserved in the correction, leading to an efficient correction from limited data and partial measurements, in this case only considering the free surface for the reconstruction.

The water and gazpacho present slightly higher errors than the other liquids in the test dataset. This is probably due to the difficulties experienced in the data acquisition and the noise of the samples in the experiment. In addition, water has higher slosh frequencies and a longer slosh time. Therefore, more data would be needed to reduce the error, even in a low-data regime. Despite this, the method already performs the adaptation on train and test information correctly.

Finally, Fig. 7 shows renders of the simulation results before and after correction compared to the snapshot that they predict in time. Additionally, the renders are complemented by a comparison between the discretized particle solution and the free surface of the ground truth. The absolute error is indicated in mm on a colored scale to visually analyze the magnitude of the distortions more accurately. It is worth mentioning that, although the particle configuration could be distorted after training to match the free surface (since it is the only information available) losing the sense of the geometry, the method manages to preserve the fluid volume configuration. The time is indicated to correlate each render to the error in Fig. 7. Despite the correction, the algorithm outputs a shape that corresponds to the real entity. As a result, giving only the free surface of the observed fluids, the networks

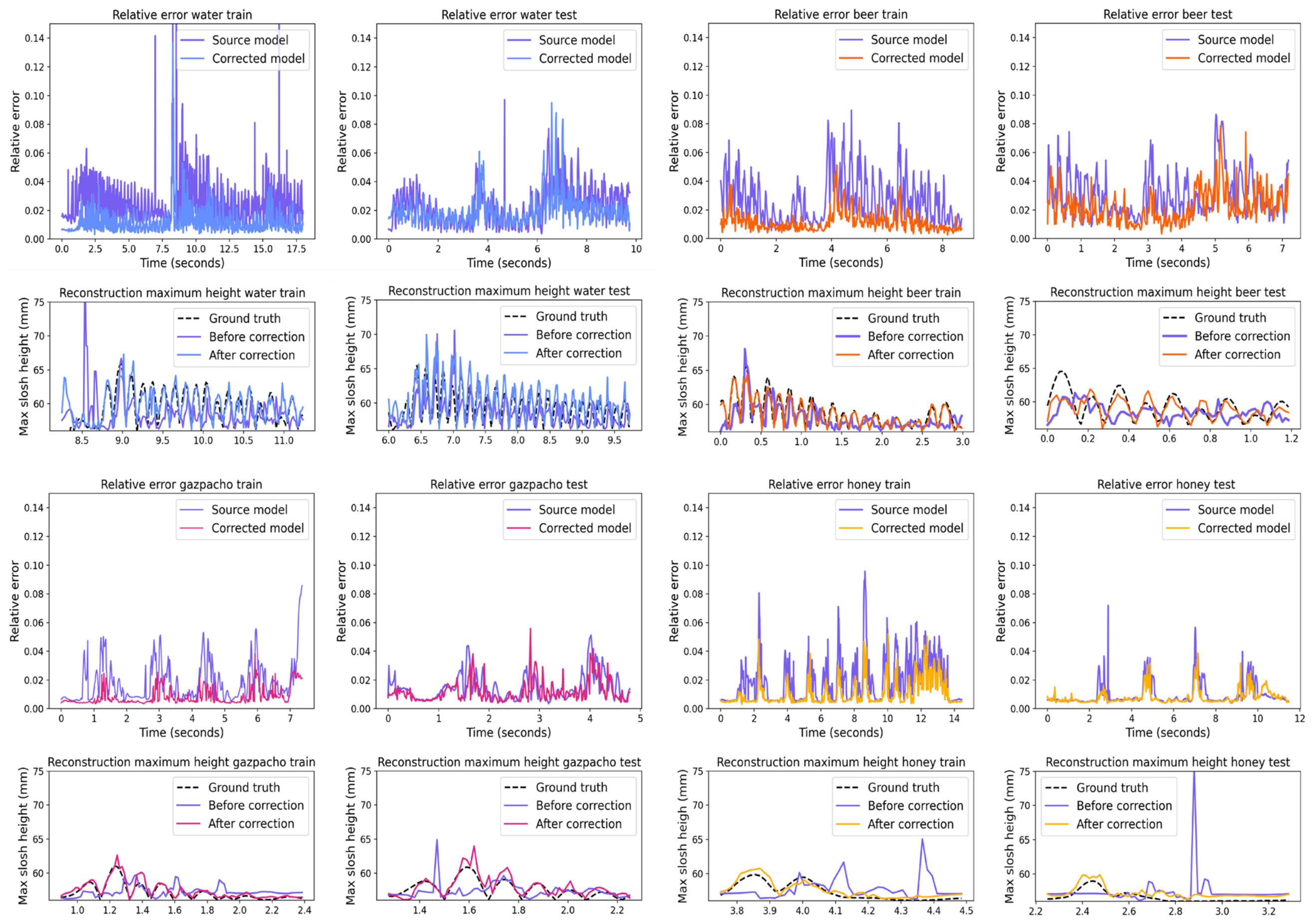

Fig. 6 Results of adaptation to a new computational liquid for beer, water, gazpacho, and honey. The relative errors of the reconstruction for the training database and the information of an unseen recording are presented. Despite the complexity and the few data employed, the correction in the test dataset is also noticeable. Relative error representations are accompanied by the maximum height reconstruction of one of the states perceived to compare the performance of before and after correction

integrate in time the dynamical evolution of the liquid and provide a three-dimensional reconstruction of the fluid volume. Particle discretization represents a fluid volume that can be translated to the user through rendering and augmented reality. This representation bridges the gap between reality and the virtual environment to provide augmented information to the machine and the user for decision-making.

7 Conclusion

The present work shows a DDDAs methodology for cognitive digital twins guided by GENERIC as an inductive bias for perception and reasoning about fluid sloshing. The algorithm learns from observations to accurately mimic new fluid behaviors from the observation of the free surface. The method provides a tool for model inference with real data from partial observations of complex dynamics. Correction of the perception of physics enables the machine to adapt and learn previously unseen liquids present in daily tasks with unknown properties. We start from a source model trained with computational data to learn a physically sound simulator of the sloshing dynamics upon GENERIC to ensure the physical consistency and generalization of the results. Calculations are performed in a low-dimensional manifold to ensure real-time performance. Thus, we obtain real-time interaction with the environment in which the model, or digital twin of the real liquid, operates to have response capacity.

We illustrate the benefits of physics-informed deep learning for correction. Given an off-line learned simulator for one particular fluid (glycerine in our case), our method manages to evolve to a new representation of the dynamics to match the behavior of previously unseen liquids. The recorded data are approximately 10 seconds. In this low-data regime, the method adapts to the new dynamics perceived. The good performance of the method can also be observed in the test recording, which has not been seen by the network before. The success is attributed to the insights learned in the source model with simulation data and the inductive bias imposed by GENERIC, which ensures the fulfillment of the principles of

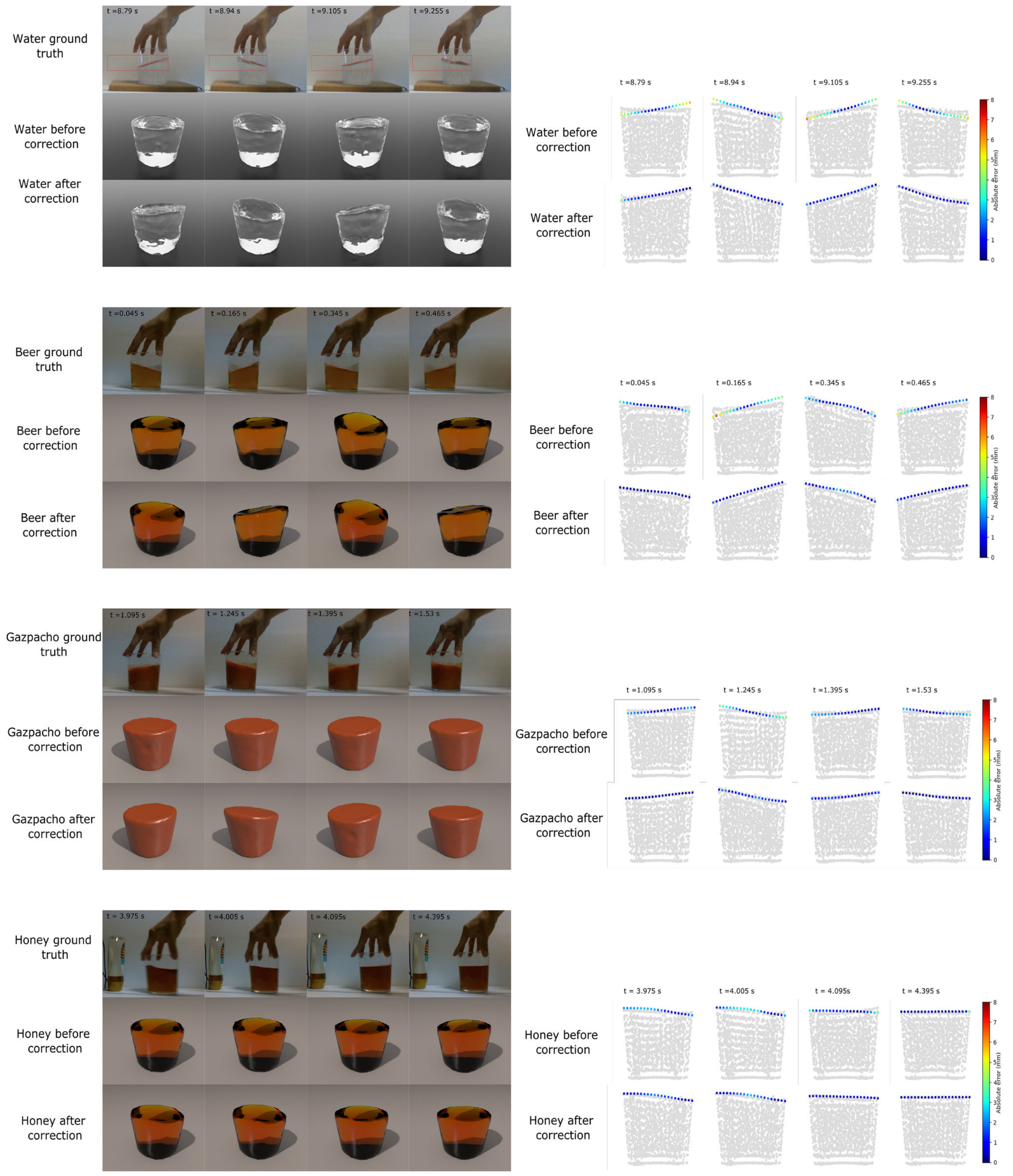

Fig. 7 Reconstruction of volume before and after correction. The algorithm outputs a discretization of the fluid particles that can be presented as a three-dimensional rendering of the volume for visualization and interpretation. The figure shows the peaks of the dynamics observed in one of the recordings

Table 2 Update parameters

	lr	wd	GRU layers selected/total	SPNN layers selected/total	λ	Epochs
Comp. water	5e−5	1e−5	1/3	4/13	2000	2000
Comp. blood	5e−5	1e−5	1/3	4/13	2000	2000
Comp. butter	5e−5	1e−5	1/3	4/13	2000	2000
Real water	1e−4	1e−5	1/3	5/13	2000	4000
Real beer	1e−4	1e−5	1/3	5/13	2000	4000
Real gazpacho	1e−4	1e−5	1/3	5/13	2000	4000
Real honey	1e−4	1e−5	1/3	5/13	2000	4000

thermodynamics. These are sufficiently general and precise to allow the simulator to evolve smoothly to new liquids. Table 2 summarizes the parameters used in the correction step.

The inductive biases already used can be complemented by other known restrictions that not only benefit the learning and correction but also the interpretability. For example, inspired by Galerkin projection methods, a continuation of this work would consider including restrictions such as orthogonality in the autoencoder to have a more interpretable reduced order space [13]. In addition, the restrictions will also boost the optimization of the learning procedure, constraining the neural networks to learning specific structures from data.

We have obtained satisfactory results using an autoencoder as a nonlinear model order reduction technique. In the next step, new neural network structures are considered to efficiently learn the correlations among the state variables and adapt to the geometry of the problem including spatial information with Convolutional and Graph neural networks.

A challenge in physics perception is the balance between adaptivity and the risk of learning noise coming from the experimental nature of the data acquisition technique. Liquids and vessels are non-Lambertian; i.e., they do not have a diffusely reflecting surface or matter, which is convenient for depth estimation. Despite the fine-tuning of the camera, the measurements include noise and invalid measurements from which the free surface has to be reconstructed. By applying transfer learning and performing slow training, we prevented the network from learning meaningless information from measurements. Moreover, the patterns already learned help reconstruct the information to learn the new behaviors with precision.

Despite the precision observed in the reconstruction of train and test recordings, the performance of the method could be further improved over the last model learned by retraining with new data sets acquired by the same means. Therefore, it can make corrections not only to persist in improving the reconstruction but also to adapt to the evolving nature of the scenario.

The observed results can be a starting point for adapting the method to new geometries and including this new parameter in the optimization from the application of geometric deep learning. Furthermore, this problem could be further extended with more general liquid detection techniques [12,41] capable of analyzing transparent and nontextured elements from different perspectives.

Acknowledgements This work has been partially funded by the Spanish Ministry of Science and Innovation, AEI /10.13039/501100011033, through Grant number PID2020-113463RB-C31 and the Regional Government of Aragon and the European Social Fund, group T24-20R. The authors also acknowledge the support of the ESI Group through the UZ-2019-0060 project.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

1. Allen KR, Smith KA, Tenenbaum JB (2020) Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. Proc Natl Acad Sci 117(47):29302–29310
2. Andrychowicz OM, Baker B, Chociej M, Jozefowicz R, McGrew B, Pachocki J, Petron A, Plappert M, Powell G, Ray A, Schneider J (2020) Learning dexterous in-hand manipulation. Int J Robot Res 39(1):3–20
3. Atkeson CG, Schaal S (1997) Learning tasks from a single demonstration. In: Proceedings of international conference on robotics and automation, vol 2. IEEE, pp 1706–1712
4. Bates CJ, Yildirim I, Tenenbaum JB, Battaglia P (2019) Modeling human intuitions about liquid flow with particle-based simulation. PLoS Comput Biol 15(7):e1007210
5. Battaglia PW, Hamrick JB, Tenenbaum JB (2013) Simulation as an engine of physical scene understanding. Proc Natl Acad Sci 110(45):18327–18332

6. Blakseth SS, Rasheed A, Kvamsdal T, San O (2022) Deep neural network enabled corrective source term approach to hybrid analysis and modeling. Neural Netw 146:181–199
7. Cai S, Liang J, Gao Q, Xu C, Wei R (2019) Particle image velocimetry based on a deep learning motion estimator. IEEE Trans Instrum Meas 69(6):3538–3554
8. Chinesta F, Cueto E, Abisset-Chavanne E, Duval JL, Khaldi FE (2020) Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. Arch Comput Methods Eng 27(1):105–134
9. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
10. Darema F (2005) Grid computing and beyond: the context of dynamic data driven applications systems. Proc IEEE 93(3):692–697
11. Darema F (2013) Dynamic data driven applications systems (dddas). Technical report, Air Force Office of Scientific Research, Arlington
12. Do C, Schubert T, Burgard W (2016) A probabilistic approach to liquid level detection in cups using an rgb-d camera. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2075–2080
13. Eivazi H, Le Clainche S, Hoyas S, Vinuesa R (2022) Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows. Expert Syst Appl 202:117038
14. Espanol P, Serrano M, Öttinger HC (1999) Thermodynamically admissible form for discrete hydrodynamics. Phys Rev Lett 83(22):4542
15. Goertzel B, Pennachin C (2007) Artificial general intelligence, vol 2. Springer
16. González D, Chinesta F, Cueto E (2019) Learning corrections for hyperelastic models from data. Front Mater 6:14
17. González D, Chinesta F, Cueto E (2019) Thermodynamically consistent data-driven computational mechanics. Contin Mech Thermodyn 31(1):239–253
18. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press
19. Goswami S, Anitescu C, Chakraborty S, Rabczuk T (2020) Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theor Appl Fract Mech 106:102447
20. Gregory SR (2018) Physical properties of glycerine. In: Glycerine. CRC Press, pp 113–156
21. Grmela M, Öttinger HC (1997) Dynamics and thermodynamics of complex fluids. I. Development of a general formalism. Phys Rev E 56(6):6620
22. Guastoni L, Güemes A, Ianiro A, Discetti S, Schlatter P, Azizpour H, Vinuesa R (2021) Convolutional-network models to predict wall-bounded turbulence from wall quantities. J Fluid Mech 928:A27
23. Hartley R, Zisserman A (2003) Multiple view geometry in computer vision. Cambridge University Press
24. Hernandez Q, Badias A, Gonzalez D, Chinesta F, Cueto E (2021) Deep learning of thermodynamics-aware reduced-order models from data. Comput Methods Appl Mech Eng 379:113763
25. Hernández Q, Badías A, González D, Chinesta F, Cueto E (2021) Structure-preserving neural networks. J Comput Phys 426:109950
26. Herschel WH, Bulkley R (1926) Konsistenzmessungen von gummi-benzollösungen. Kolloid-Z 39(4):291–300
27. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
28. Kraus M (2021) Metriplectic integrators for dissipative fluids. In: International conference on geometric science of information. Springer, pp 292–301
29. Kubo R (1966) The fluctuation-dissipation theorem. Rep Prog Phys 29(1):255
30. Laroche R, Barlier M (2017) Transfer reinforcement learning with shared dynamics. In: 31st AAAI conference on artificial intelligence
31. Lee K, Trask N, Stinis P (2021) Machine learning structure preserving brackets for forecasting irreversible processes. Adv Neural Inf Process Syst 34:5696–5707
32. Levine S, Finn C, Darrell T, Abbeel P (2016) End-to-end training of deep visuomotor policies. J Mach Learn Res 17(1):1334–1373
33. Liu CK, Negrut D (2021) The role of physics-based simulators in robotics. Ann Rev Control, Robot, Auton Syst 4:35–58
34. Mohebujjaman M, Rebholz LG, Iliescu T (2019) Physically constrained data-driven correction for reduced-order modeling of fluid flows. Int J Numer Methods Fluids 89(3):103–122
35. Monaghan JJ (1992) Smoothed particle hydrodynamics. Ann Rev Astron Astrophys 30:543–574
36. Moya B, Badías A, Alfaro I, Chinesta F, Cueto E (2020) Digital twins that learn and correct themselves. Int J Numer Methods Eng 123(13):3034–3044
37. Moya B, Badias A, Gonzalez D, Chinesta F, Cueto E (2021) Physics perception in sloshing scenes with guaranteed thermodynamic consistency. arXiv preprint arXiv:2106.13301
38. Rabault J, Kolaas J, Jensen A (2017) Performing particle image velocimetry using artificial neural networks: a proof-of-concept. Meas Sci Technol 28(12):125301
39. Richter F, Orosco RK, Yip MC (2009) Thermodynamically consistent time-stepping algorithms for non-linear thermomechanical systems. Int J Numer Methods Eng 79(6):706–732
40. Richter F, Orosco RK, Yip MC (2022) Image based reconstruction of liquids from 2d surface detections. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13811–13820
41. Sajjan S, Moore M, Pan M, Nagaraja G, Lee J, Zeng A, Song S (2020) Clear grasp: 3d shape estimation of transparent objects for manipulation. In 2020 IEEE Int Confer Robot Autom (ICRA). IEEE, pp 3634–3642
42. Sanchez-Gonzalez A, Godwin J, Pfaff T, Ying R, Leskovec J, Battaglia P (2020) Learning to simulate complex physics with graph networks. In: International conference on machine learning. PMLR, pp 8459–8468
43. Schenck C, Fox D (2018) Perceiving and reasoning about liquids using fully convolutional networks. Int J Robot Res 37(4–5):452–471
44. Schenck C, Fox D (2017) Reasoning about liquids via closed-loop simulation. arXiv preprint arXiv:1703.01656
45. Schenck C, Fox D (2018) Spnets: differentiable fluid dynamics for deep neural networks. In: Conference on robot learning. PMLR, pp 317–335
46. Smith M. Approximate viscosities of some common liquids
47. Wiewel S, Becher M, Thuerey N (2019) Latent space physics: Towards learning the temporal evolution of fluid flow. In: Computer graphics forum, vol 38. Wiley Online Library, pp 71–82
48. Zhang H, Nguyen H, Bui XN, Pradhan B, Asteris PG, Costache R, Aryal J (2021) A generalized artificial intelligence model for estimating the friction angle of clays in evaluating slope stability using a deep neural network and Harris Hawks optimization algorithm. Eng Comput 1–14
49. Zhang Z, Shin Y, Em Karniadakis G (2021) Gfinns: generic formalism informed neural networks for deterministic and stochastic dynamical systems. arXiv preprint arXiv:2109.00092