Computational Mechanics
https://doi.org/10.1007/s00466-024-02564-3

ORIGINAL ARTICLE

A comparison of single and double generator formalisms for thermodynamics-informed neural networks

Pau Urdeitx[1] · Icíar Alfaro[1] · David González[1] · Francisco Chinesta[2,3] · Elías Cueto[1]

Received: 27 March 2024 / Accepted: 23 September 2024

Abstract
The development of inductive biases has been shown to be a very effective way to increase the accuracy and robustness of neural networks, particularly when they are used to predict physical phenomena. These biases significantly increase the certainty of predictions, decrease the error made and allow considerably smaller datasets to be used. There are a multitude of methods in the literature to develop these biases. One of the most effective ways, when dealing with physical phenomena, is to introduce physical principles of recognised validity into the network architecture. The problem becomes more complex without knowledge of the physical principles governing the phenomena under study. A very interesting possibility then is to turn to the principles of thermodynamics, which are universally valid, regardless of the level of abstraction of the description sought for the phenomenon under study. To ensure compliance with the principles of thermodynamics, there are formulations that have a long tradition in many branches of science. In the field of rheology, for example, two main types of formalisms are used to ensure compliance with these principles: one-generator and two-generator formalisms. In this paper we study the advantages and disadvantages of each, using classical problems with known solutions and synthetic data.

Keywords Thermodynamics-informed neural networks · Scientific machine learning · GENERIC · Single generator formalism

1 Introduction

Since the recent re-emergence of machine learning after some "artificial intelligence winters", with neural networks and deep learning as major players, there has been a growing interest in constraining or controlling such learning, moving from "black box" learning to "grey box" learning for scientific machine learning purposes [1–3]. When learning physical phenomena, the predictability and accuracy of the results become a major requisite. The imposition of certain mathematical or data structures, which allow us to establish inductive biases on the learned systems, has given rise to the development of different families of neural networks capable of learning the physical evolution of a system from the data to a great accuracy. Among them, Physics-Informed Neural Networks (PINNs) stand out, in which the learning algorithm tries to fit the solution to a known equation, defined by the governing partial differential equation, from the data [4–8]. Taking advantage of the symmetries in the data, seen from a thermodynamic perspective, and imposing a specific, well-known structure on the evolution of its state variables of a dynamic system, the Structure Preserving Neural Networks (SPNN) and Hamiltonian Neural Networks (HNN), among others, can be found in the literature [9–12]. The application of structures or formalisms during the learning process allows conservation laws (system symmetries) to be learned

✉ Elías Cueto
ecueto@unizar.es

Pau Urdeitx
purdeitx@unizar.es

Icíar Alfaro
iciar@unizar.es

David González
gonzal@unizar.es

Francisco Chinesta
francisco.chinesta@ensam.eu

[1] ESI Group-UZ Chair of the National Strategy on Artificial Intelligence. Aragon Institute of Engineering Research (I3A) Universidad de Zaragoza, María de Luna, s.n., 50018 Zaragoza, Spain

[2] ESI Group Chair at the PIMM Lab, Arts et Mètiers Institute of Technology, 151 Bvd. de l'Hôpital., 75013 Paris, France

[3] CNRS@CREATE LTD., 1 CREATE Way, Singapore 138602, Singapore

Published online: 19 October 2024

without direct supervision. These methods, which employ inductive biases, have been shown not only to be able to learn the dynamics of complex systems, ensuring the fulfillment of basic laws of thermodynamics but also these constraints can improve the robustness of the method during inference, limiting the appearance of incoherent responses. A review of the evolution of the integration of known physics—particularly, thermodynamics—in neural networks can be found in [13].

These restrictions, however, do not always represent an advantage during learning. There is a trade-off between the expressiveness of a network, i.e., its ability to model complex functions, and the learnability of a network, the capability of a machine learning model to acquire knowledge or improve its performance from data [14]. Completely unrestricted (black box) networks represent the maximum expressiveness of the network but are often unable to capture the underlying physics of the problem for previously unseen situations, while by incorporating inductive biases, the representativeness increases at the cost of compromising the learning process.

From a physics perspective, different approaches have been presented for the representation of the evolution of a dynamical system out of equilibrium [15]. At the molecular dynamics level, for instance, Newton's laws (or their equivalent Hamiltonian or Lagrangian alternatives) are sufficient for the description of the system. At this scale, everything is reversible or conservative. However, this entails the control of position and momentum of a number of molecules of the order of the Avogadro number at each instant of time, being unfathomable except in very specific cases. As we move from the microscopic to a meso and macroscopic description of the system, tracking the state variables becomes intricate, since unresolved variables play a role in the evolution of the physics, thus introducing the dependence on history [16, 17]. This lack of information is generally associated with a corrective term that allows us to go from an ideal system (reversible) to a real system (irreversible) which is associated with the generation of entropy of the system. In this sense, the total energy of the system is conserved, as dictated by the first law of thermodynamics.

A convenient way to define the dynamics of non-equilibrium systems is through a generalization of the Poisson bracket with its extension for irreversible systems with the so-called dissipative bracket [18–20]. This way of defining systems compacts the main properties (invariants) by carefully defining the operators as well as considering Casimir invariants as constraints in the system [21]. The choice of one set of variables or another to represent the system can give rise to different bracket formalisms derived from this generalization of the Hamiltonian for non-conservative systems [22–24].

The level of compaction in the description of complex dynamical systems together with the preservation of the mathematical and thermodynamic properties of the system, make these formalisms ideal for learning systems through neural networks [12]. In this sense, structure-preserving models have demonstrated better performance regarding the use of black boxes, reinforcing the hypothesis of the benefit of considering these structures to shape inductive biases in data-driven models [7, 25, 26].

This paper analyzes the learning of different physical phenomena through the imposition of two alternative formalisms, with a thermodynamically consistent structure, widely used in the field of rheology, among other fields: single generator bracket and double generator bracket [27]. Recently, more elaborated formalisms of this type have been proposed that employ a 4-bracket formalism, but these, in general, will hinder the learning process [28]. Both formalisms are mathematical structures for the description of dynamics in non-equilibrium systems, in which reversible mechanics is described by Hamilton's principle of least action and irreversible dynamics is a bilinear dissipation term [22, 23]. The difference between the two structures analyzed lies in the definition of the energy generator functionals. While the single generator formalism defines a single generator, metriplectic systems, such as GENERIC, are defined with two generators, one associated with the reversible dynamics and the other with the dissipative part [19, 29]. Although there are correlations between both formulations, and the transformation from one formalism to another can be obtained theoretically, the consideration of a generator or two establishes key differences that can be relevant during learning processes. Thus, the objective of this paper is to examine these differences and the advantages and limitations they confer to develop structure-preserving Neural Network systems. The pros and cons of each of the formulations will be analysed by considering two different problems, one discrete and one continuous, after the appropriate discretisation by means of finite elements. The different parameters affecting the learning process are analysed in detail and conclusions are drawn on the advantages and disadvantages of each method.

2 Methods

2.1 Single and double generator bracket formalisms

In 1984 different authors presented distinct, although very similar in spirit, formulations for irreversible phenomena as an extension of the classical Hamiltonian approach [18–20]. For instance, [21] starts by considering that equilibrium is achieved by extremizing the energy at constant entropy,

$$\mathcal{F}_\lambda = \mathcal{H} + \lambda \mathcal{S}, \tag{1}$$

being $\mathcal{F}$ the generalized free energy of the system, λ a Lagrange multiplier, and where $\mathcal{H}$ and $\mathcal{S}$, correspond to the

Hamiltonian, and the entropy of the system, respectively. In general, one form of introducing dissipation in the Hamiltonian description of a system is by adding a Casimir or generalized entropy functional. Casimirs are functionals that are conserved for all Hamiltonians. Therefore, by considering Casimirs, we can drop the Lagrange multiplier to arrive at a very convenient description of the free energy of our system in the form

$$\mathcal{F} = \mathcal{H} + \mathcal{S}. \tag{2}$$

Based on the generalized free energy, different descriptions of the energy can be proposed which can be used to define different bracket formalisms [11, 27, 30]. If we consider a single energy potential, $\mathcal{F}$, the free energy in the system, the equations of motion for a set of state variables z will be:

$$\frac{dz}{dt} = \{\{z, \mathcal{F}\}\}, \tag{3}$$

where the double braces are employed to denote a dissipative generalization of the Poisson bracket. Since any operator can be split into the self-adjoint and anti-self-adjoint parts, we arrive at

$$\frac{dz}{dt} = \{\{z, \mathcal{F}\}\} = \{z, \mathcal{F}\} + [z, \mathcal{F}]. \tag{4}$$

By considering the description of the brackets [18], this system can be written in the algebraic form with two operators, $\boldsymbol{L}$ and $\boldsymbol{M}$ as:

$$\boldsymbol{L} : T^*\mathcal{M} \to T\mathcal{M}, \boldsymbol{M} : T^*\mathcal{M} \to T\mathcal{M}, \tag{5}$$

being $T^*\mathcal{M}$, and $T\mathcal{M}$ the cotangent and tangent bundles of $\mathcal{M}$, respectively. Equation (3) can thus be rewritten as [18]:

$$\{\{z, \mathcal{F}\}\} = \boldsymbol{L}\frac{\partial \mathcal{F}}{\partial z} + \boldsymbol{M}\frac{\partial \mathcal{F}}{\partial z}. \tag{6}$$

The operator $\boldsymbol{L}$ is the symplectic or Poisson matrix—it defines a Poisson bracket—, which is defined as a skew-symmetric matrix. The operator $\boldsymbol{M}$ is the dissipative matrix, defined as a positive semi-definite matrix [31].

By decomposing Eq. (6) into the Hamiltonian or conservative energy, $\mathcal{H}$, and the entropy, $\mathcal{S}$, a two-generator bracket can be defined as:

$$\frac{dz}{dt} = \{z, (\mathcal{H} + \mathcal{S})\} + [z, (\mathcal{H} + \mathcal{S})], \tag{7}$$

which also can be written as:

$$\frac{dz}{dt} = \boldsymbol{L}\frac{\partial(\mathcal{H} + \mathcal{S})}{\partial z} + \boldsymbol{M}\frac{\partial(\mathcal{H} + \mathcal{S})}{\partial z}. \tag{8}$$

To ensure (i) conservation of the total energy $d\mathcal{H}/dt = 0$, and (ii) non-negative entropy production $d\mathcal{S}/dt \geq 0$, two additional conditions must be fulfilled. Based on the definition of Casimir invariants $\mathcal{C}$, [32]:

$$\frac{\partial \mathcal{C}}{\partial \boldsymbol{x}}\boldsymbol{J}\frac{\partial \mathcal{F}}{\partial \boldsymbol{x}} = \boldsymbol{0}, \forall \mathcal{F}, \tag{9}$$

to ensure energy conservation we must impose that:

$$\frac{\partial \mathcal{H}}{\partial z}\boldsymbol{M} = \boldsymbol{0}, \tag{10}$$

and, equivalently, to ensure non-negative entropy production,

$$\frac{\partial \mathcal{S}}{\partial z}\boldsymbol{L} = \boldsymbol{0}. \tag{11}$$

In this way, it is straightforward to prove that the conservation of energy is obtained through

$$\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial z}\left(\boldsymbol{L}\frac{\partial \mathcal{H}}{\partial z} + \boldsymbol{M}\frac{\partial \mathcal{S}}{\partial z}\right) = 0, \tag{12}$$

provided that

$$\frac{\partial \mathcal{H}}{\partial z}\boldsymbol{L}\frac{\partial \mathcal{H}}{\partial z} = 0.$$

In turn, non-negative entropy production results from

$$\frac{d\mathcal{S}}{dt} = \frac{\partial \mathcal{S}}{\partial z}\left(\boldsymbol{L}\frac{\partial \mathcal{H}}{\partial z} + \boldsymbol{M}\frac{\partial \mathcal{S}}{\partial z}\right) \geq 0, \tag{13}$$

given that

$$\frac{\partial \mathcal{S}}{\partial z}\boldsymbol{M}\frac{\partial \mathcal{S}}{\partial z} \geq 0,$$

by the semi-positive definiteness of the matrix $\boldsymbol{M}$.

By imposing these two degeneracy conditions on the double generator formalism we thus arrive at the so-called "General Equation for Non-Equilibrium Reversible-Irreversible Coupling" formalism, GENERIC, as [31]:

$$\frac{\partial z}{\partial t} = \{z, \mathcal{H}\} + [z, \mathcal{S}]. \tag{14}$$

Although the equivalence between the two formalisms is demonstrated theoretically, the choice of one or the other formalism has practical implications in the development of methods that have to determine the particular structure of the equations of motion of our system from data. In essence, both formalisms will need to determine the particular form of the matrices $\boldsymbol{L}$ and $\boldsymbol{M}$, but one will also need to determine the form of a single potential, see Eq. (4), while the second formalism uses two, see Eq. (14). In the latter case, it will be necessary to explicitly require the fulfillment of the degeneracy conditions by defining Casimirs of $\boldsymbol{L}$, and $\boldsymbol{M}$ in

their construction. This can be done due to the separation of energy and entropy. While this can be done analytically in both formalisms, due to the methodology of the training algorithm, these Casimirs are not enforced in the single generator. This is likely to result in a method with a more general structure, but subject to more constraints, and which will have the undeniable advantage of the explicit imposition of the two principles of thermodynamics (conservation of energy, non-negative production of entropy) [22].

It is also well-known that a Poisson bracket must also satisfy the so-called Jacobi identity [29]. While this is obviously true, this condition is in general extremely difficult to guarantee a priori even for analytical developments. When applied to neural networks architectures, on the contrary, it may result in methods that have strong difficulties to learn. In our previous works, we have demonstrated that it is very often preferable to simple avoid an explicit imposition of this condition [10, 31].

Our problem will then be defined as finding the precise form of the evolution of the state variables of the system, z, from experimental measurements on the system, given predetermined initial conditions, $z(0)$:

$$\dot{z} = \frac{dz}{dt} = f(z,t), \boldsymbol{x} \in \Omega \in \mathbb{R}^D, t \in \mathcal{I} = (0,T], z(0) = z_0, \tag{15}$$

where $\boldsymbol{x}$ are the spatial coordinates on a domain Ω, and $f(\boldsymbol{x}, z, t)$ being the function that describes the flow map $z_0 \rightarrow z(z_0, T)$ in a prescribed time horizon T. The use of inductive biases will consist precisely in assuming that the precise form of the function f sought will be either Eq. (4) or Eq. (14).

For this purpose, both formalisms will be discretised in time, so that

$$z(t+\Delta t) = z_t + \left(\boldsymbol{L}^S + \boldsymbol{M}^S\right)\frac{\partial \mathcal{F}}{\partial z}\Delta t, \tag{16}$$

where S refers to the single generator bracket formalism, and:

$$z(t+\Delta t) = z_t + \left(\boldsymbol{L}^G\frac{\partial \mathcal{H}}{\partial z} + \boldsymbol{M}^G\frac{\partial \mathcal{S}}{\partial z}\right)\Delta t, \tag{17}$$

where G stands for the double generator bracket or GENERIC, for short, formalism. These will be omitted when there is no risk of confusion. Note that the scheme in Eq. (16) resembles closely the so-called OnsagerNet [11]. In that case, however, the system is assumed to be close to equilibrium and matrix $\boldsymbol{M}$ is assumed to be constant. It also includes an autoencoder in its architecture so as to unveil the latent variables governing the problem. This approach is also present in [10], but has not been considered here in order to keep the analysis as simple as possible.

2.2 Thermodynamics-Informed Neural Networks

Neural networks are well known to satisfy the universal approximation theorem, so the time evolution of the state variables, assumed in the form given by Eq. (4) or by Eq. (14), will be determined using feed-forward neural networks (Fig. 1). The input of the network is formed by the state vector of the system at each instant of time, $z(\boldsymbol{x}, t)$, while the output of the net is taken as the parameters necessary for the reconstruction of the integration formalisms, including $\boldsymbol{L}(\boldsymbol{x}, t)$, and $\boldsymbol{M}(\boldsymbol{x}, t)$ operators, and the energy generators, $\mathcal{F}$, in the single generator formalism, and $\mathcal{H}$ and $\mathcal{S}$, in the GENERIC formlism. $\boldsymbol{L}$, the symplectic matrix, is well-known to be skew-symmetric. Therefore, it is more convenient to learn a matrix $\boldsymbol{l}$, such that $\boldsymbol{L} = \boldsymbol{l} - \boldsymbol{l}^T$. Conversely, $\boldsymbol{M}$, the friction matrix, is symmetric and positive semi-definite, so it is more convenient to learn a matrix $\boldsymbol{m}$ such that $\boldsymbol{M} = \boldsymbol{m}\boldsymbol{m}^T$ [31]. Moreover, the gradient of the potentials is computed from the learned energy (scalar) by using the `autograd` function of PyTorch.

The state of the variables of the system in the next time step, $z_{n+1} = z(\boldsymbol{x}, t+\Delta t) = z(\boldsymbol{x}, (n+1)\Delta t)$, is then obtained by the integration with the reconstructed formalism, the current state of variables, z_n, and a fixed time step increment, Δt.

The loss function includes up to three contributions depending on the integration formalism. The first term of the loss function, the data loss, $\mathcal{L}_n^{\text{data}}$, enforces the agreement of the predicted values of the variables to the reference values. The data loss, $\mathcal{L}_n^{\text{data}}$, compares the mean square error between the predicted values, z_n^{net}, and the ground truth values, z_n^{GT}, throughout the time series, by employing the L2 norm.

$$\mathcal{L}_n^{\text{data}} = \| \, z_{n+1}^{\text{GT}} - z_{n+1}^{\text{net}} \, \|_2^2 \, . \tag{18}$$

A second term of the loss, the degeneracy loss $\mathcal{L}_n^{\text{degen}}$, enforces the fulfillment of the degeneracy conditions,

$$\mathcal{L}_n^{\text{degen}} = \| \, \boldsymbol{L}\nabla\mathcal{S} \, \|_2^2 + \| \, \boldsymbol{M}\nabla\mathcal{H} \, \|_2^2, \tag{19}$$

A third term of the loss, the regularization loss, $\mathcal{L}^{\text{reg}}$, is considered to avoid the overfitting of the network, which is defined as the sum of the squared weight parameters of the network.

$$\mathcal{L}^{\text{reg}} = \sum_{l}^{L}\sum_{j}^{n^{[l]}}\sum_{i}^{n^{[l+1]}} (w_{i,j}^{[l]})^2, \tag{20}$$

where l is the index of the current network layer and $\boldsymbol{w}^{[l]}$ is the weight matrix of this same layer. $n^{[l]}$ represents the number of neurons at layer $l \in [1, L]$.

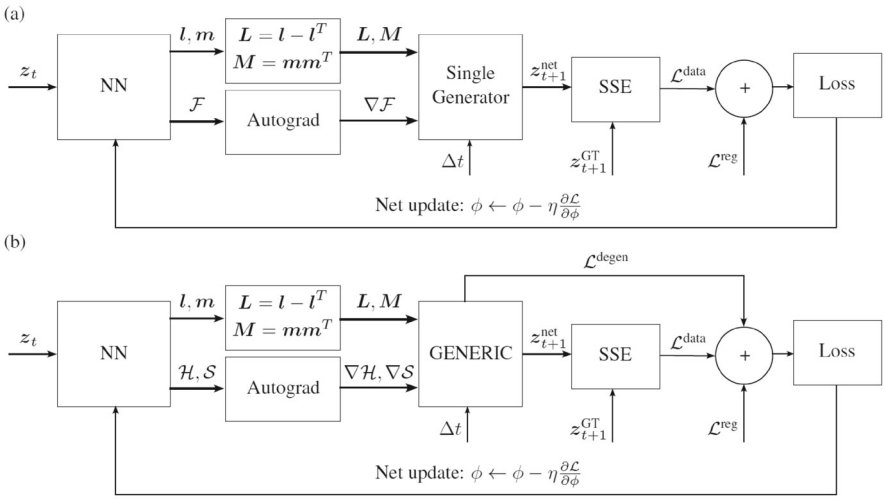

Fig. 1 Configuration scheme of the Structure Preserving Neural Network (SPNN) employed to learn single generator **a** and GENERIC **b** formalisms. The input parameters of the net are the state of the system, $z(x, t)$, at each time step. The output of the net includes the energy $\mathcal{F}$, $\mathcal{H}$, and $\mathcal{S}$, as well as the m, and l components needed to reconstruct the formalism. The integration of the formalism gives the state of the system at the next time step $z(x, t+1)$. Then, the Data error and the degeneracy conditions are computed to define the loss of the net

Then, the global loss function is the sum of the contributions of the loss functions just considered. Due to the differences in the magnitude of each term in the loss, compensation weights were considered for the data and regularization losses,

$$\mathcal{L} = \sum_{n=1}^{N_T} \left(\lambda_d \mathcal{L}_n^{\text{data}} + \mathcal{L}_n^{\text{degen}} \right) + \lambda_r \mathcal{L}^{\text{reg}}, \tag{21}$$

where λ_d, and λ_r were the data, and the regularization weight compensation hyperparameters, respectively. N_T represents the number of snapshots in each simulation.

Based on the loss thus obtained, the parameters of the net (weights and biases) are updated through the backpropagation algorithm with the gradient descent technique. This process is repeated for the fixed number of epoch $\mathtt{n}_{\text{epoch}}$, with a multistep learning rate scheduler with a decaying factor in $1/3\mathtt{n}_{\text{epoch}}$, and $2/3\mathtt{n}_{\text{epoch}}$.

The training database was composed of the time series ($t \in \mathcal{I}(0, T]$) of a collection of different trajectories of the dynamic systems. The trajectories of the database were divided into training ($N_{\text{train}} = 80\%$ of the dataset) and test data ($N_{\text{test}} = 20\%$ of the dataset).

The performance of the network is evaluated based on the predicted state of the variables of the system by comparing them with the ground truth values by calculating the mean square error (MSE) for all trajectories, and throughout all the time series, for every variable in the problem,

$$\text{MSE}^{\text{data}}(z) = \frac{1}{N_T} \sum_{n=1}^{N_T} \left(z_n^{\text{GT}} - z_n^{\text{net}} \right)^2. \tag{22}$$

The pseudocode of the training and test methods for the single generator bracket are presented in Algorithm 1 and 2, respectively, while the pseudocode of the training and test for the GENERIC formalism can be seen in Algorithm 3 and 4, respectively.

Algorithm 1 Pseudocode for the training algorithm of the single generator bracket SPNN

Load train database: z^{GT}(train partition), Δt;
Initialize w_i, b_i;
for epoch $\leftarrow$ 1 to $\mathtt{n}_{\text{epoch}}$ **do**
  for train case $\leftarrow$ 1 to N_{train} **do**
    Initialize state vector: z_0^{net} is z_0^{GT};
    Initialize losses: $\mathcal{L}^{\text{data}} = 0$;
    for snapshot $\leftarrow$ 1 to N_T **do**
      Forward propagation: $[\boldsymbol{l}, \boldsymbol{m}, \mathcal{F}] \leftarrow \text{Net}(z_t^{\text{GT}})$;
      Take the Energy gradient (autograd of PyTorch): $\nabla\mathcal{F} \leftarrow \nabla_z\mathcal{F}$;
      Formalism construction: $\boldsymbol{L} \leftarrow \boldsymbol{l} - \boldsymbol{l}^T$, $\boldsymbol{M} \leftarrow \boldsymbol{m}.\boldsymbol{m}^T$;
      Time integration $z_{t+1}^{\text{net}} \leftarrow z_t^{\text{net}} + \Delta t(\boldsymbol{L} + \boldsymbol{M})\nabla\mathcal{F}$;
      Update data loss $\mathcal{L}^{\text{data}} \leftarrow \mathcal{L}^{\text{data}} + \mathcal{L}_n^{\text{data}}$;
      Update degeneracy loss $\mathcal{L}^{\text{degen}} \leftarrow \mathcal{L}^{\text{degen}} + \mathcal{L}_n^{\text{degen}}$;
    end for
    SSE loss function: $\mathcal{L} \leftarrow \lambda_d \mathcal{L}^{\text{data}} + \lambda_r \mathcal{L}^{\text{reg}}$
    Backward propagation;
    Optimizer step;
  end for
  Learning rate scheduler;
end for

Algorithm 2 Pseudocode for the test algorithm of the single generator bracket SPNN

Load test database: z^{GT}(test partition), Δt;
Load network parameters;
for test case $\leftarrow$ 1 to N_{test} **do**
  Initialize state vector: z_0^{net} is z_0^{GT};
  for snapshot $\leftarrow$ 1 to N_T **do**
    Forward propagation $[\boldsymbol{l}, \boldsymbol{m}, \mathcal{F}] \leftarrow \text{Net}(z_t^{\text{GT}})$;
    Formalism construction: $\boldsymbol{L} \leftarrow \boldsymbol{l} - \boldsymbol{l}^T$, $\boldsymbol{M} \leftarrow \boldsymbol{m} \cdot \boldsymbol{m}^T$;
    Take the Energy gradient (autograd of PyTorch): $\nabla\mathcal{F} \leftarrow \nabla_z\mathcal{F}$;
    Time integration $z_{t+1}^{\text{net}} \leftarrow z_t^{\text{net}} + \Delta t(\boldsymbol{L} + \boldsymbol{M})\nabla\mathcal{F}$;
  end for
  Compute MSE;
end for

Algorithm 3 Pseudocode for the training algorithm of the GENERIC SPNN

```
Load train database: z^GT(train partition), Δt;
Initialize w_i, b_i;
for epoch ← 1 to n_epoch do
  for train case ← 1 to N_train do
    Initialize state vector: z_0^net is z_0^GT;
    Initialize losses: L^data, L^deg = 0;
    for snapshot ← 1 to N_T do
      Forward propagation: [l, m, H, S] ← Net(z_t^GT);
      Take the Energy gradient (autograd of PyTorch): ∇H ← ∇_z H, ∇S ← ∇_z S;
      Formalism construction: L ← l − l^T, M ← m · m^T;
      Time integration z_{t+1}^net ← z_t^net + Δt(L∇H + M∇S);
      Update data loss L^data ← L^data + L_n^data;
      Update degeneracy loss L^degen ← L^degen + L_n^degen;
    end for
    SSE loss function: L ← λ_d L^data + L^degen + λ_r L^reg
    Backward propagation;
    Optimizer step;
  end for
  Learning rate scheduler;
end for
```

Algorithm 4 Pseudocode for the test algorithm of the GENERIC SPNN

```
Load test database: z^GT(test partition), Δt;
Load network parameters;
for test case ← 1 to N_test do
  Initialize state vector: z_0^net is z_0^GT;
  for snapshot ← 1 to N_T do
    Forward propagation [l, m, H, S] ← Net(z_t^GT);
    Formalism construction: L ← l − l^T, M ← m · m^T;
    Take the Energy gradient (autograd of PyTorch): ∇H ← ∇_z H, ∇S ← ∇_z S;
    Time integration z_{t+1}^net ← z_t^net + Δt(L∇H + M∇S);
  end for
  Compute MSE;
end for
```

3 Numerical results

3.1 Double thermoelastic pendulum

3.1.1 System description

The first example considers a double thermoelastic pendulum, which consists of two masses, m_1 and m_2, connected by two springs with natural lengths λ_1^0, and λ_2^0 (Fig. 2) [33]. This model includes thermal effects due to the Gough-Joule effects, including the heat flux between springs (dissipative dynamics), and movements of masses (Hamiltonian mechanics). The set of variables which describe the system are:

$$S = \{\boldsymbol{Z}(x,t) = (\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{p}_1, \boldsymbol{p}_2, s_1, s_2) \in (\mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R})\}, \quad (23)$$

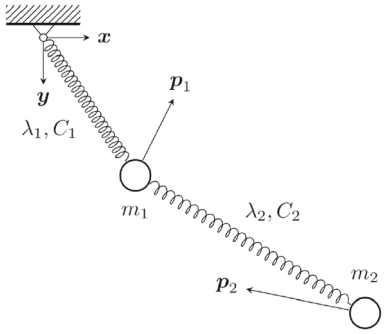

Fig. 2 Double thermo-elastic pendulum system

being, $\boldsymbol{q}_i$, $\boldsymbol{p}_i$, and s_i the position, linear momentum, and entropy of each mass of the system.

The total energy of the system is defined by the sum of the kinetic energy of the masses, K_i, and the internal energy of the springs, e_i, as:

$$E(\boldsymbol{z}) = \sum_i \left(K_i(\boldsymbol{z}) + e_i(\lambda_i, s_i)\right), \quad (24)$$

being λ_i defined by the position of the masses as:

$$\lambda_1 = \sqrt{\boldsymbol{q}_1 \cdot \boldsymbol{q}_1}, \ \lambda_2 = \sqrt{(\boldsymbol{q}_2 - \boldsymbol{q}_1) \cdot (\boldsymbol{q}_2 - \boldsymbol{q}_1)}, \quad (25)$$

and the kinetic energy, K_i, as:

$$K_i = \frac{1}{2m_i} |\boldsymbol{p}_i|^2. \quad (26)$$

3.1.2 Net hyperparameters and database

A thermodynamically consistent algorithm following [33] and implemented in MATLAB has been employed to generate the training database. The parameters associated with the generation of the synthetic data include the weights of the masses $m_1 = 1$ kg and $m_2 = 2$ kg, the natural lengths of the springs, $\lambda_1 = 2$ m and $\lambda_2 = 1$ m, and the thermal constant $C_1 = 0.02$ J and, $C_2 = 0.2$ J, of the first and second pendulum, respectively (see Fig. 2). The conductivity is $\kappa = 300$ and the simulation time is $T = 60$ s in time increments of $\Delta t = 0.3$ s ($N_t = 200$ snapshots). The database, state vector $\boldsymbol{Z}(x,t)$, Eq. (23), contains $N_x = 50$ different trajectories, split randomly into 40 train and 10 test trajectories. Each trajectory has been obtained with mean initial conditions around the initial position of $\boldsymbol{q}_1 = [4.5, 4.5]^T$ m, and $\boldsymbol{q}_2 = [2.0, 4.5]^T$ m, and initial momentum of $\boldsymbol{p}_1 = -0.5, 1.5]^T$ kg.m/s, and $\boldsymbol{p}_2 = [1.4, -0.2]^T$ kg.m/s (variations of 5% around the mean initial position of $\boldsymbol{q}_1$, and $\boldsymbol{p}_1$ have been simulated for 20 s to take the initial positions).

The net is composed of an input layer, $N_{\text{in}} = 10$, and an output layer, whose size depends on the chosen formalism,

as $N_{\text{out}}^G = N_{\text{in}}^2 + 2 = 102$ and $N_{\text{out}}^S = N_{\text{in}}^2 + 1 = 101$, for GENERIC and single generator bracket, respectively. The number of hidden layers in both cases is $N_{\text{hidden}} = 5$ with softplus function activation and with $N_h = 2N_{\text{in}}^2 = 200$ units of neurons each. It is initialized according to the Kaiming method [34], with normal distribution, and the optimizer used is Adam [35], with a weight decay of $\lambda_r = 10^{-5}$ and data loss weight of $\lambda_d = 10^2$. A total number of epochs of $N_{\text{epoch}} = 12000$, with a multistep learning rate scheduler, is used, starting in $\mu = 10^{-4}$ and decaying by a factor of $\gamma = 0.1$ in epochs 4000, and 8000 ($1/3.N_{\text{epoch}}$, and $2/3.N_{\text{epoch}}$, respectively). The evolution of the terms of data loss, $\mathcal{L}^{\text{data}}$, and degeneracy loss, $\mathcal{L}^{\text{degen}}$, have been represented in Fig. 3.

3.1.3 Results

The state variables z_n^{net}, obtained at each time increment n, from the reconstruction of the system with each formalism show a good degree of coherence with the synthetic ground truth data, z_n^{GT} (see Figs. 4, 5). In both cases, the entropy variables show the highest error during the reconstruction. The comparison between the errors obtained in the reconstruction of the data does not show a significant difference between both formalisms (Fig. 6a). In the ground truth case, the error obtained in the train by the single generator formalism is lower than that obtained by the GENERIC formalism. However, GENERIC shows less error in the reconstruction of the test, previously unseen trajectories.

To compare the thermodynamic consistency of both formalisms, the energy reconstructed with the predicted values of the state variables, $\mathcal{H}(z_n^{\text{net}})$, with the GENERIC and single generator formalisms are compared to the real energy calculated from the system variables, $\mathcal{H}(z_n^{\text{GT}})$) through Eq. (24). The error for the theoretical energy of the system has been represented in Fig. 6b.

3.2 Couette flow of an Oldroyd-B fluid

3.2.1 System description

The second example is an Oldroyd-B fluid within a shear flow (Couette flow), which can be modeled as a viscoelastic fluid composed of a series of linear elastic dumbbells (representing, for instance, the effect of polymer chains) immersed in a solvent [36, 37]. The dynamics of this system can be obtained from both, a macroscopic and microscopic perspective. The chosen set of variables to describe the system are (Fig. 7):

$$\mathcal{D} = \{\boldsymbol{z}(y,t) = (\boldsymbol{q}, \boldsymbol{v}, e, \tau) \in (\mathbb{R}^2 \times \mathbb{R} \times \mathbb{R} \times \mathbb{R})\}, \tag{27}$$

being, $\boldsymbol{q}$, and $\boldsymbol{v}$, the position and velocity vectors, e, the internal energy, and, τ the stress-shear component of the conformal tensor.

These parameters arise from the solution of the problem in two different scales. The macroscopic solution of the problem can be obtained through the Fokker-Plank equation, by applying the CONNFFESSIT technique, and by its transformation into the Itô stochastic differential equation as [38, 39]:

$$dq_x = \left(\frac{\partial \boldsymbol{v}}{\partial y} q_y - \frac{1}{2\text{We}} q_x\right) dt + \frac{1}{\sqrt{\text{We}}} dV_t, \tag{28}$$

$$dq_y = -\frac{1}{2\text{We}} q_y dt + \frac{1}{\sqrt{\text{We}}} dW_t, \tag{29}$$

being $\boldsymbol{v} = [v_x, v_y]^T$, and $\boldsymbol{q} = [q_x, q_y]^T$, the velocity and position vectors, We, the Weissenberg number, and V_t and W_t are two independent one-dimensional Brownian motions. Under the assumption of the Couette Flow, the dependencies of the positions are given by $q_x = q_x(y,t)$ and $q_y = q_y(t)$. The solution of this equation can be obtained by Monte Carlo techniques, considering the empirical mean to replace the mathematical expectation.

In the microscopic scale, the evolution of the conformation tensor $\boldsymbol{c} = \langle \boldsymbol{r}\boldsymbol{r} \rangle$, describes the state of the dumbbells through the expected τ_{xy} component of this tensor. This acts as an internal variable in the system and is given by:

$$\tau_{xy} = \frac{\epsilon}{\text{We}} \frac{1}{K} \sum_{k=1}^{K} q_x q_y, \tag{30}$$

being $\epsilon = \frac{\nu_p}{\nu_s}$, the ratio of the polymer to solvent viscosities, and K the number of dumbbells in the simulation.

For its part, the viscoelastic behavior of the model can be defined by the mechanical model of the solvent (s), as a Newtonian substrate, and polymer chains (p), as linear dumbbells diluted in the substrate. Thus, the deviatoric part, $\boldsymbol{T}$, of the stress tensor, $\boldsymbol{\sigma}$, is defined as:

$$\boldsymbol{T} + \lambda_1 \overset{\nabla}{\boldsymbol{T}} = \eta_0 \left(\dot{\boldsymbol{\gamma}} + \lambda_2 \overset{\nabla}{\dot{\boldsymbol{\gamma}}}\right), \tag{31}$$

being, λ_1, λ_2, and, η_0, parameters of the model, $\dot{\boldsymbol{\gamma}}$ is the strain rate tensor, given by $\dot{\boldsymbol{\gamma}} = (\nabla^s \boldsymbol{v}) = \boldsymbol{D}$, and where the triangle denotes non-linear Oldroyd upper convected derivative [36].

Then, the total stress is given by the stress in the solvent (s) and polymer (p), given by:

$$\boldsymbol{T} = \eta_s \dot{\boldsymbol{\gamma}} + \boldsymbol{\tau}, \tag{32}$$

so that

$$\boldsymbol{\tau} + \lambda_1 \overset{\nabla}{\boldsymbol{\tau}} = \eta_p \dot{\boldsymbol{\gamma}}, \tag{33}$$

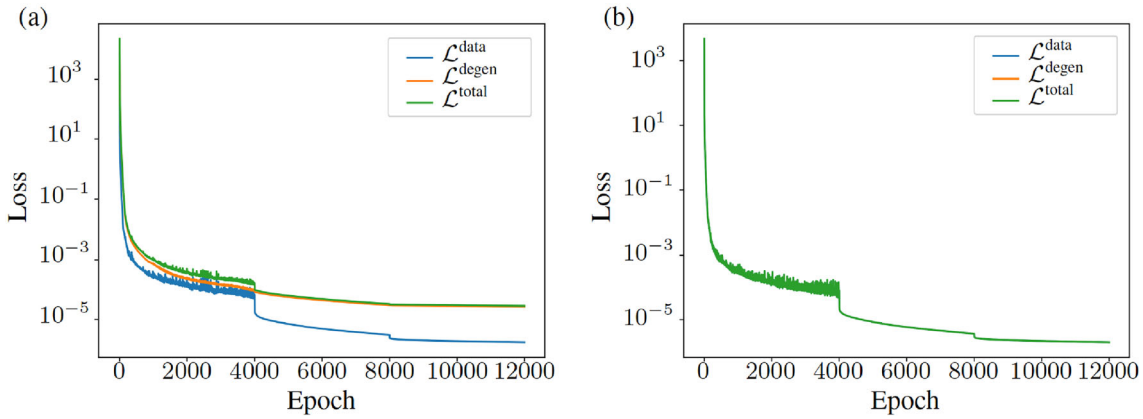

Fig. 3 Loss representation in training for GENERIC **a** and single generator **b** formalisms (Log scale). The loss function of the GENERIC includes data loss, $\mathcal{L}^{\text{data}}$, and degeneracy loss $\mathcal{L}^{\text{degen}}$, while the only contribution on the single generator is the data loss, $\mathcal{L}^{\text{data}}$

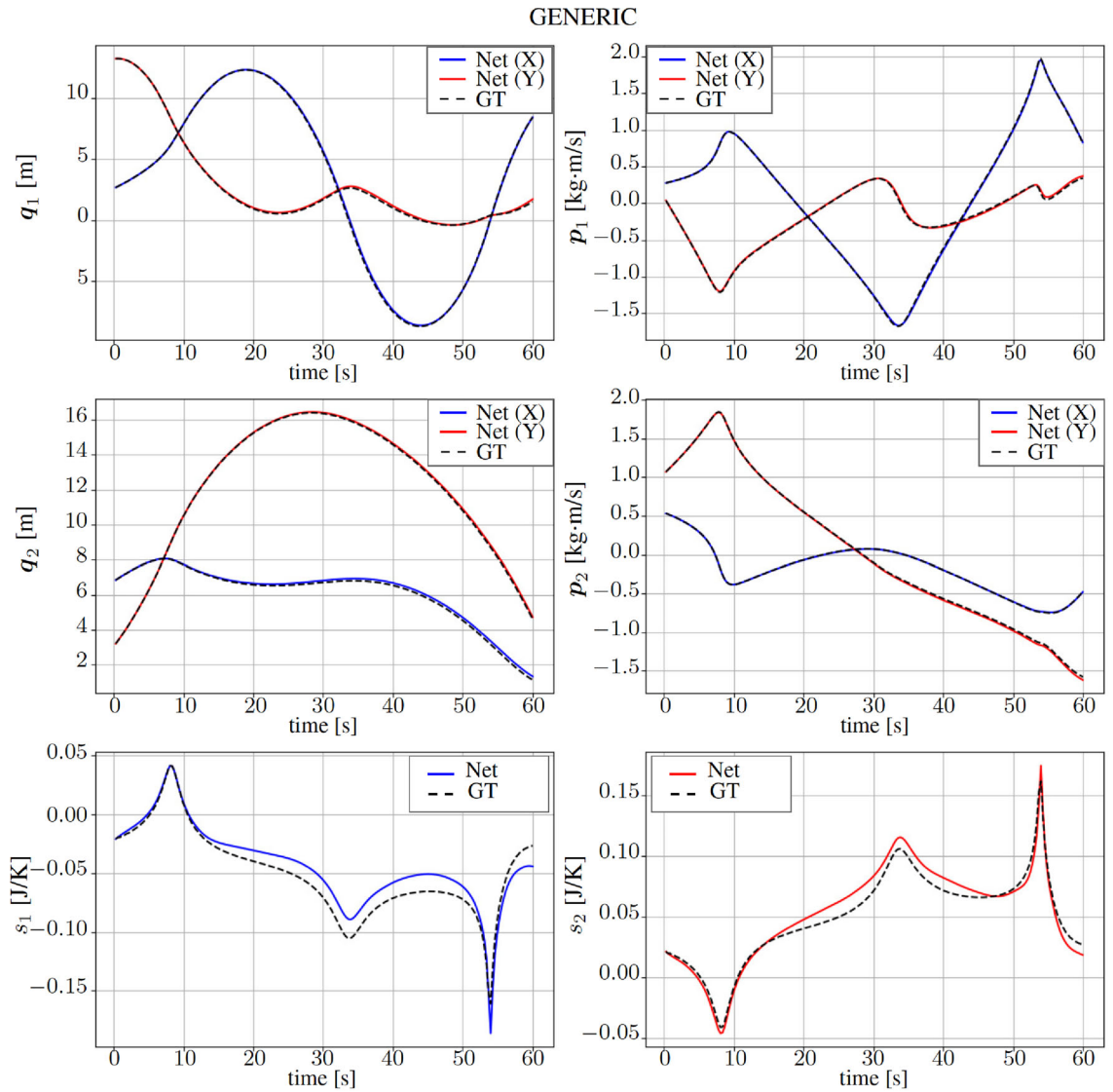

Fig. 4 Results of the reconstruction of the double thermo-elastic pendulum system. Test trajectory (Ground Truth, GT) and the reconstruction of the double thermo-elastic pendulum using time integration with GENERIC formalism

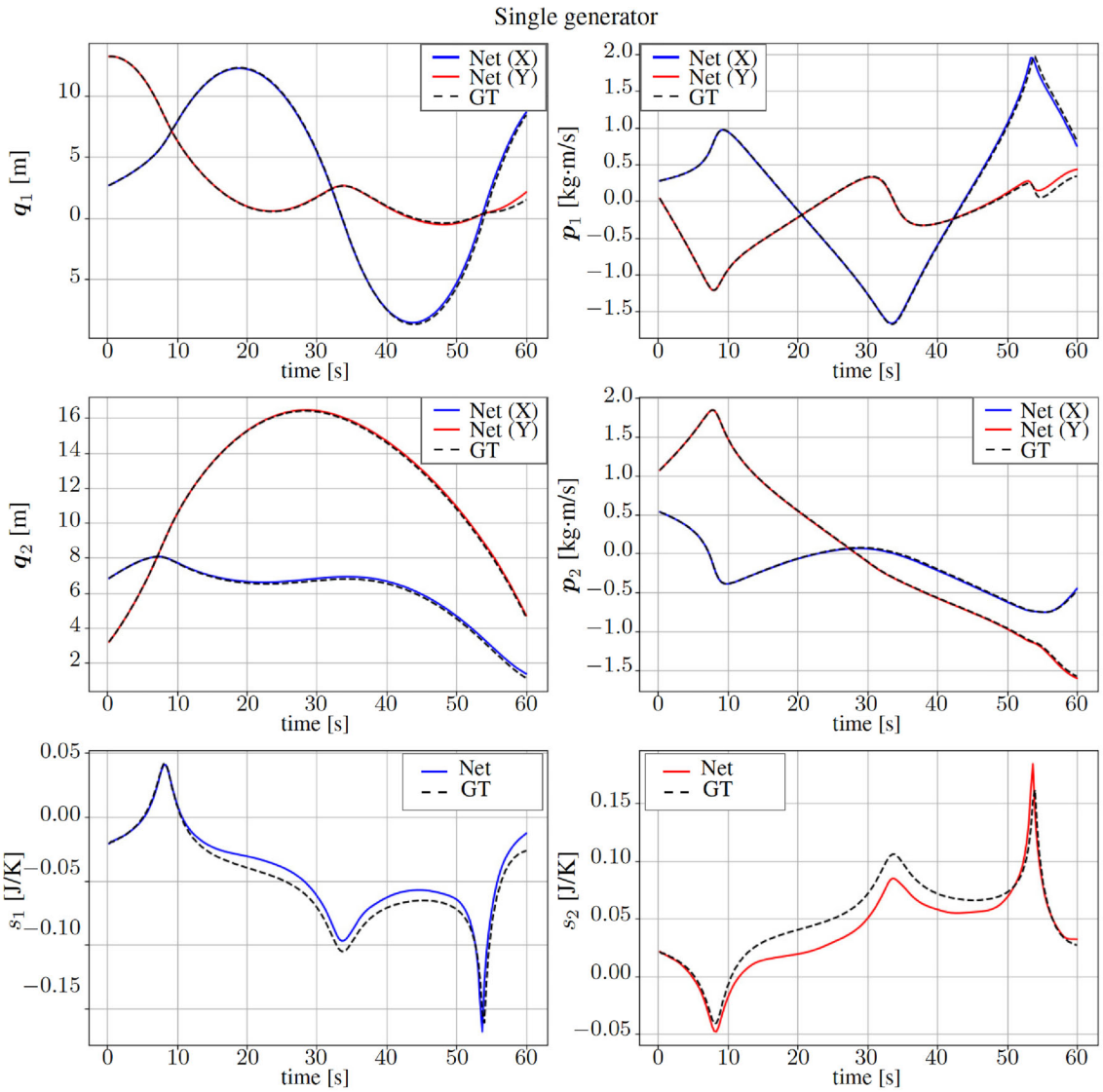

Fig. 5 Results of the reconstruction of the double thermo-elastic pendulum system. Test trajectory (Ground Truth, GT) and the reconstruction of the double thermo-elastic pendulum using time integration with single generator formalism

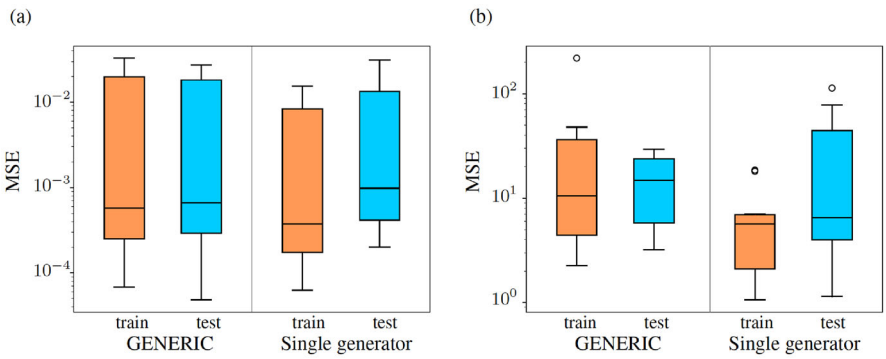

Fig. 6 Results of the Net. **a** MSE error in the system variables, obtained by the single generator and GENERIC formalisms. **b** Error in the reconstruction of the energy ($\mathcal{H}$) of the train and test trajectories calculated from the system variables

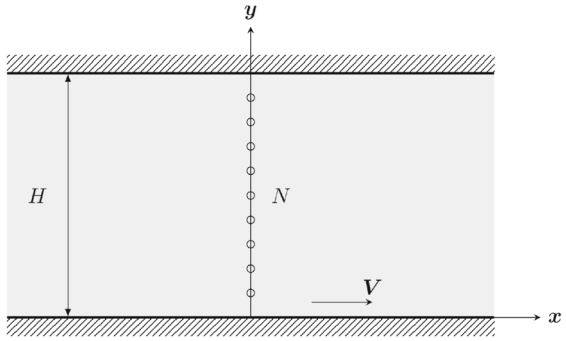

Fig. 7 Couette flow in an Oldroyd-B fluid

which is the constitutive equation of the elastic stress.

3.2.2 Net hyperparameters and database

A dimensionless multi-scale algorithm implemented in MATLAB has been employed to generate the training database. The parameters associated with the generation of the synthetic data include the lid velocity, $V = 1$ m/s, the number of dumbbells at each node of the model, $N_d = 10000$, the number of Reynolds, Re=0.1, and Weissenberg, We= 1.0. The fluid vertical direction has been discretized with $N_x = 100$ nodes, and the simulation time is $T = 1$ s in time increments of $\Delta t = 0.0067$ s ($N_t = 150$ snapshots). At the node at $h = H$ a condition of no-slip conditions has been imposed ($v = 0$ m/s), therefore, it has been excluded from the database. The database state vector $\boldsymbol{z}(y, t)$, Eq. (27), contains $N_z = 100$ trajectories, split into 80 training and 20 test trajectories.

The net is composed of an input layer, $N_{\text{in}} = 5$, and an output layer, depending on the training Formalism, of $N^G_{\text{out}} = N^2_{\text{in}} + 2 = 27$ and $N^S_{\text{out}} = N^2_{\text{in}} + 1 = 26$, for GENERIC and single generator formalisms, respectively. The number of hidden layers in both cases is $N_{\text{hidden}} = 5$ with Softplus function activation and with $N_h = 2N^2_{\text{in}} = 50$ units of neurons each. It is initialized according to the Kaiming method [34], with normal distribution, and the optimizer used is Adam [35], with a weight decay of $\lambda_r = 10^{-5}$ and data loss weight of $\lambda_d = 10^2$. A total number of epochs of $N_{\text{epoch}} = 6000$, with a multistep learning rate scheduler, is used, starting in $\mu = 10^{-4}$ and decaying by a factor of $\gamma = 0.1$ in epochs 2000 and 4000 ($1/3.N_{\text{epoch}}$, and $2/3.N_{\text{epoch}}$, respectively). The evolution of the terms of data loss, $\mathcal{L}^{\text{data}}$, and degeneracy loss, $\mathcal{L}^{\text{degen}}$, has been represented in Fig. 8.

3.2.3 Results

The evolution of the state variables of the model, reconstructed by the GENERIC, and single generator, has been represented in Fig. 9a, b, respectively. Even though the capacity of the net is lower than in the previous example, the obtained reconstruction shows at least one order of magnitude less error than in the previous system (Fig. 10). The results show a lower error for the reconstruction of the state variables in the single generator than in the GENERIC case. Likewise, a lower error is observed in the test than in the train, which can be attributed to specific trajectories close to the limits ($x = 0$). The MSE error in the energy (internal energy of the system, e) shows better performance of the single generator than in the GENERIC formalism (Fig. 10b).

4 Discussion

The inductive bias introduced when learning the different systems through the enforcement of well-known physics formalisms introduces interesting characteristics that are worth commenting on. We would like to point out that a comparison with a black box algorithm has not been made, since this has already been done with the GENERIC formalism in [31]. In all the cases studied, GENERIC improved the results by about an order of magnitude.

In the example of the thermo-elastic double pendulum, both formalisms show very similar system reconstruction errors, with less error obtained by GENERIC. On the contrary, in the Oldroyd-B fluid example, the single generator formalism reports the best results. These differences can be attributed to the differences between the two systems analyzed. Since the double pendulum is a chaotic system, the trajectories within the database can be much more different from each other than those observed in the Couette flow.

By comparing the energy of the system, computed from the reconstructed variables of the system ($\mathcal{H}(z^{\text{net}}_n)$), with the ground truth energy ($\mathcal{H}(z^{\text{GT}}_n)$), less error has been observed in the single generator than in the GENERIC case, even without the conservation of energy being explicitly restricted, see Fig. 6b.

For comparison and application of these formalisms in learning the structure of a system through neural networks, it is important to understand the advantages and limitations they may present. The single generator structure, with fewer restrictions than GENERIC, has therefore a higher expressiveness, while the GENERIC structure, by the separation of the energy generators, is more representative, in the sense that it allows for an explicit imposition of the laws of thermodynamics. However, this comes at the cost of including a new hyperparameter in the loss function. This hyperparameter can increase the weight to one part of the problem or another: reconstruction of the data, or imposition of thermodynamic correctness. In this sense, forcing a very strict imposition of thermodynamics can make it difficult to find the solution to the problem, which, in part, responds to the

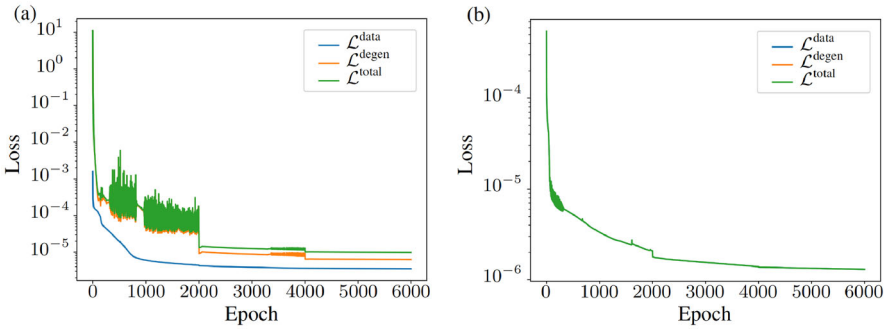

Fig. 8 Loss representation for the Couette flow in the Oldroyd Fluid system, for GENERIC **a**, including data loss, $\mathcal{L}^{\text{data}}$, and degeneracy loss $\mathcal{L}^{\text{degen}}$, and single generator **b** whose only contribution is the data loss, $\mathcal{L}^{\text{data}}$ (Log scale)

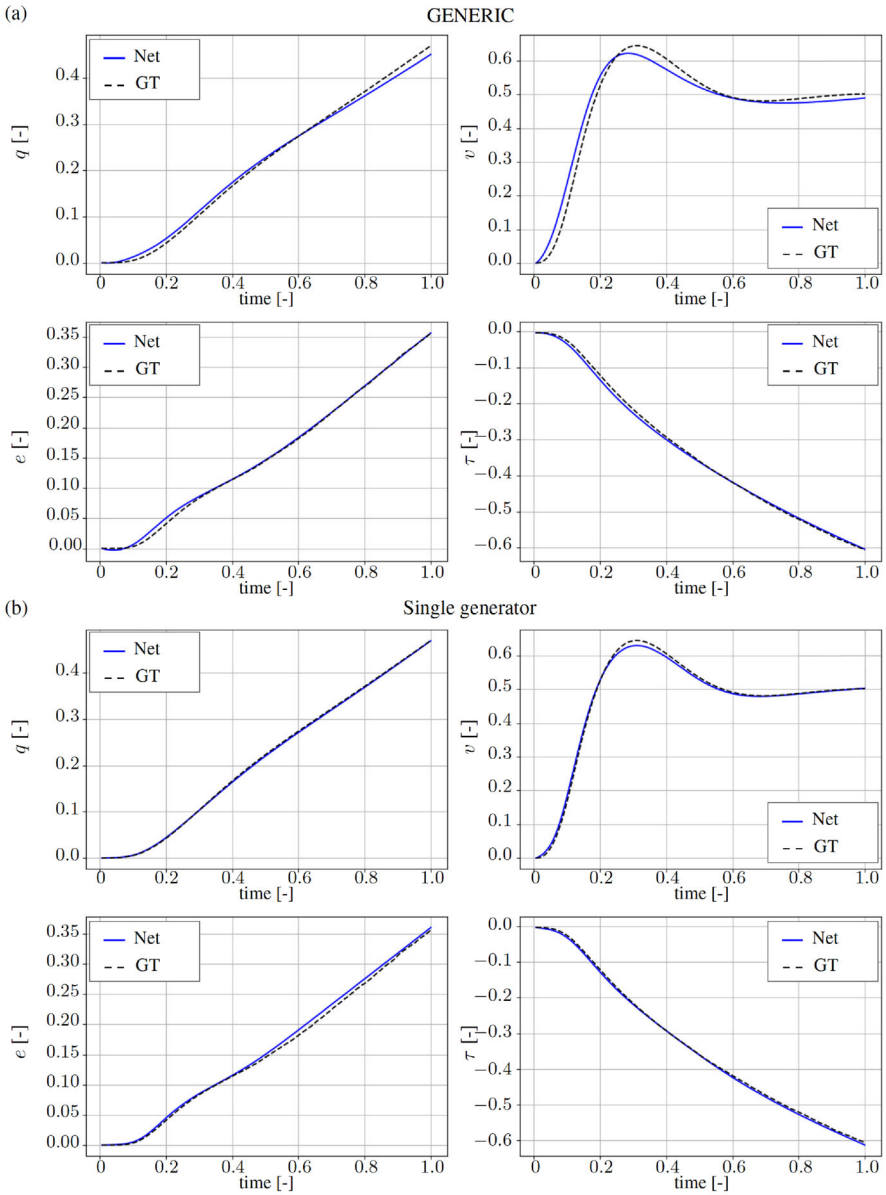

Fig. 9 Test trajectory reconstruction of the Couette flow in an Oldroyd fluid with GENERIC **a** and single generator bracket **b** formalisms

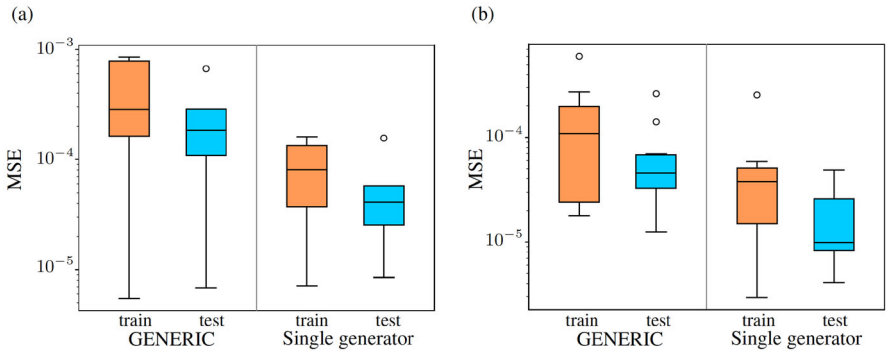

Fig. 10 Box plots for the data integration mean square error (MSE) **a**, and the energy (Hamiltonian) mean square error **b**, for the Couette flow in an Oldroyd fluid

observed advantage of the single generator over GENERIC. On the contrary, the degeneracy conditions can be seen as additional information available to the network to find the solution space for the problem.

Due to the simple time integration scheme (forward Euler), small errors obtained in the integration of the variables cause a magnification of the successive error. Thus, by increasing the number of snapshots in the database, the error of the reconstruction of the state variables increases, see Figs. 11a and 12a. Besides, by increasing the number of trajectories in the database, the generalization of the net is improved which decreases the error of the reconstruction of the state variables, see Figs. 11b and 12b. Moreover, we observed that an increase in the number of train trajectories promotes thermodynamic consistency in GENERIC, see Appendix 5 and Fig. 16.

For the case of the Couette flow in the Oldroyd-B fluid, the variation of the *Re* and *We* numbers have been studied. As *Re* and *We* change, the relative importance of the dissipative and reversible dynamics of the example is varied. It has been observed that the behavior of both formalisms differs, with GENERIC showing better behavior for weakly dissipative dynamics (Appendix 5, Fig. 17). Thus, GENERIC seems to be more advisable with dynamics including low dissipative content, while the single generator scheme shows a clear advantage with higher dissipative dynamics.

The results obtained with both formalisms are dependent on the hyperparameters of the network. The value of the learning rate seems to be the most critical parameter for the GENERIC formalism. With values of $lr \geq 1e-3$, for the case of the double thermo-elastic pendulum, the network can stagnate in a local minimum for which the dissipative component is the trivial solution $\boldsymbol{M} = \boldsymbol{0}$. For its part, the single generator seems to show a higher dependence on the network capacity and the training process, being unable to reconstruct some trajectories (test, but also train) with a low capacity, and a low number of training epochs. At the same time, its dependence on the data to generalize the solution means that, as the number of training trajectories in the database is reduced, the single generator error increases and vice versa. Thus, GENERIC shows more stable behavior and higher robustness to the conditions of the database and the hyperparameters of the network.

5 Conclusion

Both formalisms show high accuracy in the results, being coherent with the laws of thermodynamics. Even when considering a single generator, where the imposition of these is not explicit, the energy, computed by post-processing, shows a good degree of conservation. Although the single generator formalism indeed seems to yield better results in general terms, with lower computing costs (less training time), the stability of the predictions depends largely on the adequate adjustment of the network hyperparameters. Thus, the effect of the different hyperparameters, including learning rate, number of training epochs, and neurons in hidden layers, as well as differences in the database have been compared in the reconstruction of the system with both formalisms. In this sense, decreasing the capacity of the network (fewer units per hidden layer) limits the network's ability to generalize and prevents the reconstruction of some trajectories with the single generator formalism. Besides, reducing the learning rate to $lr = 1e-4$ within the single generator paradigm requires increasing the number of training epochs to be able to reconstruct the solution. This is not observed in the GENERIC formalism, which, maybe supported by the degeneracy conditions, shows higher robustness in the reconstruction of the trajectories in every tested example.

For its part, the structure in the GENERIC formalism is more representative since it separates the dynamics into two independent terms. This can be seen as an advantage but also can imply some limitations. As the energy generators were defined separately, the thermodynamic laws can be imposed explicitly by the degeneracy conditions. However, this implies the addition of an extra term on the loss function, which needs to be compensated with the data loss with an extra hyperparameter λ_d. An additional limitation of the GENERIC formalism is derived from the separation

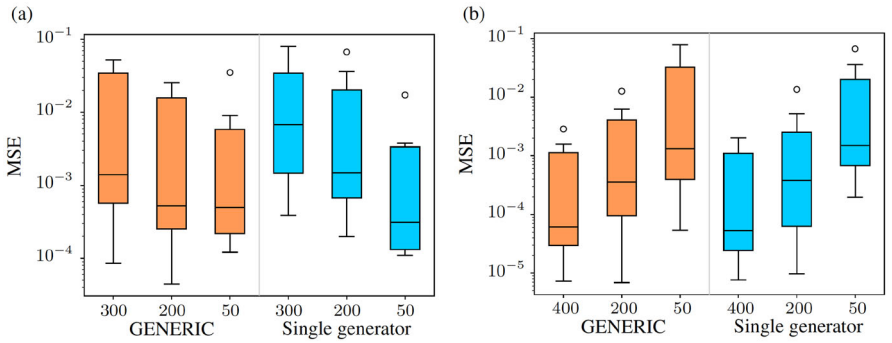

Fig. 11 Box plots for the test data integration mean square error (MSE) for the double thermo-elastic pendulum. **a** Increasing the number of snapshots (decrease in the step time) increases data error which is associated with the integration method. **b** Increasing the number of trajectories in the database improves the generalization of the network which highly reduces the error

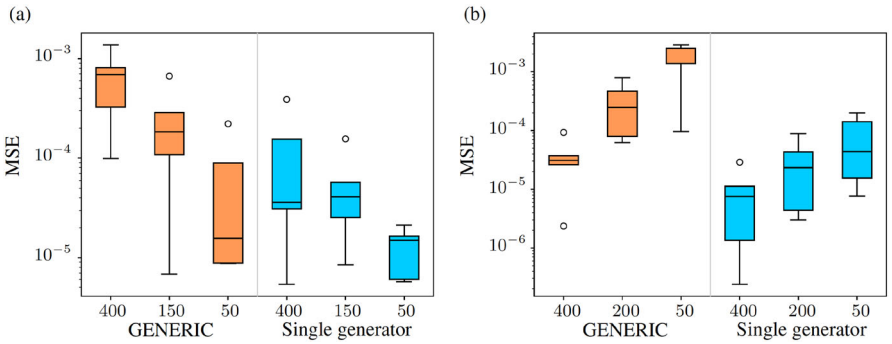

Fig. 12 Box plots for the test data integration mean square error (MSE) for the Couette flow in an Oldroyd-B fluid. **a** By increasing the number of snapshots (300, 200, and 50 snapshots), the error obtained increases. **b** By increasing the number of points (400, 200, and 50 discretized points), the error obtained significantly decreases

of energy generators. This separation into two independent terms can guide the learning process to a solution in which one of these energy potential generators is no longer relevant (trivial solution with $\boldsymbol{M}\nabla\mathcal{S} = \mathbf{0}$) and the entire weight of the dynamics of the problem is learned through the other term of the formalism.

Some of these conclusions depend on the adjustment of various hyperparameters of the network and the physics of the particular example studied. In this work, we have tried to simplify all this content to focus on a clear answer to the main question: single generator or GENERIC for learning physics? In this sense, there is no definitive winner with conclusive arguments. Both formalisms seem to be equivalent—something already demonstrated in the literature—. The addition of degeneracy conditions acts as an additional data term to obtain the solution space of the system. Moreover, the degeneracy conditions provide higher robustness to the model, allowing for a better generalization and lower errors with fewer data samples, in chaotic systems, and lower network capacity (number of neurons). For its part, the single generator formalism has shown to be capable of learning, to a certain extent, these implicit restrictions in the system, reporting generally less error in the system variables reconstruction, with less computational cost, but with a high dependence on the adjustment of the network's hyperparameters. Thus, as was exposed by B. Edwards et al. in their analysis of complex fluids, even though both formalisms can reconstruct the dynamics of different complex systems, the description of the energy through the double generator with the GENERIC structure (separated Hamiltonian and entropy) is more natural and reports some benefits in the description of dissipative dynamics [22, 23].

Appendix A. Influence of hyperparameter values

The effect of different hyperparameters in the reconstruction of the system with the GENERIC and single generator formalisms has been analyzed. The single generator shows a high sensibility on the hyperparameters of the net as

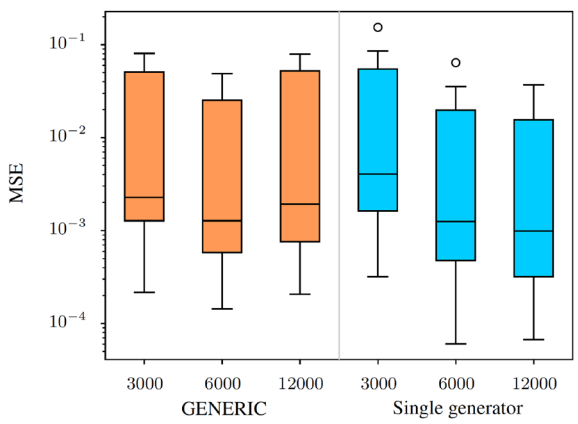

Fig. 13 Box plots for the data test integration mean square error (MSE). Effect of increasing the number of training epochs in GENERIC and single generator bracket formalisms

compared with GENERIC. Low capacity networks and insufficient training epochs make the single generator approach to be unable to reconstruct the state variables of the double thermo-elastic pendulum. For its part, the thermodynamic consistency of GENERIC shows a stronger dependence on the choice of an appropriate learning rate value. We detail the effect of these values in the following sections.

Training epochs

Generally speaking, as the number of training epochs is increased, the reported error of the net is reduced. This tendency, on the limit, will contribute to the net overfitting, which reduces the generalization of the forecasts and increases the error in the test trajectories. We have compared the effect of the increase of the maximum training epochs in the double pendulum example, by training both formalisms

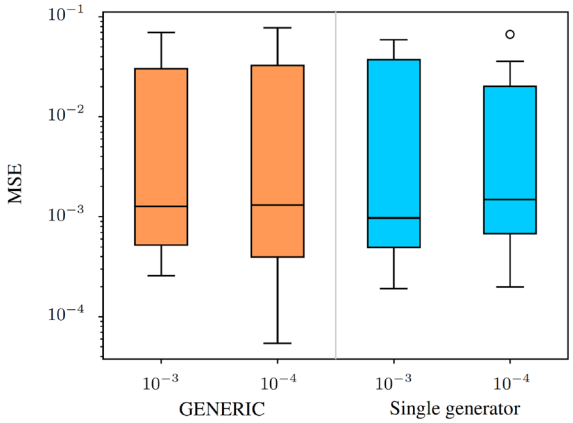

Fig. 14 Box plots for the data integration mean square error (MSE), with initial learning rates of $lr = 1e-3$, and $lr = 1e-4$ for the GENERIC and single generator bracket formalisms

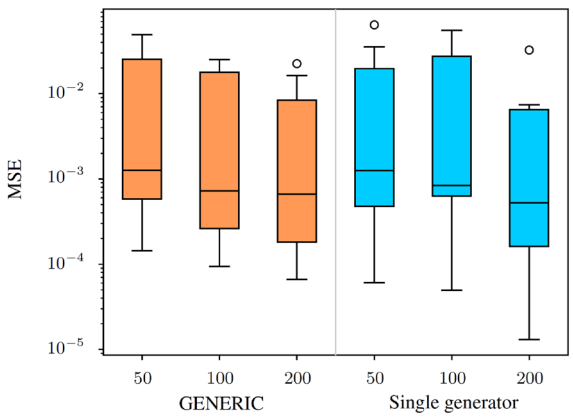

Fig. 15 Box plots for the data test integration mean square error (MSE) in the double thermo-elastic pendulum system. Effect of the net capacity, number of neurons on hidden layers, for GENERIC and single generator bracket formalisms. As the capacity of the net increases, the MSE on the reconstructed trajectories is reduced

with $N_{\text{epoch}} = 3000$, $N_{\text{epoch}} = 6000$, and $N_{\text{epoch}} = 12000$. As observed in Fig. 13, in general, for both formalisms, the error on the state variables reconstruction is reduced as the number of training epochs increases. However, for the maximum number of epochs considered in GENERIC, the error is increased. As observed in detail in the reconstruction of the model by Eq. (17), the increase of the maximum epoch moves the solution to a local minimum (trivial solution with $\boldsymbol{M} = \boldsymbol{0}$) which reduces the generalization of the problem and increases the test error. This seems to indicate that the system solution encounters a "cliff" point, and the use of a high learning rate causes it to systematically move away from the solution. If we look at Fig. 3, we can see a first region where the term associated with the data in the loss function decreases continuously but fluctuates in the term associated with the degeneracy conditions, which denotes a complicated balance between these two contributions in the Loss function. This problem is solved by decreasing the learning rate to $lr = 1e-4$.

Learning rate

The effect of different learning rates has been studied in the reconstruction of the double pendulum system with the GENERIC and single generator formalisms (Fig. 14). The error in the variables reconstruction represented shows a worsening in the error for both formalisms, much more significant in GENERIC. Then, the single generator with a learning rate of $lr = 1e-3$ reports a lower error than GENERIC. Additionally, for the same learning rate value, GENERIC has reported falling into a local minimum (trivial solution $M = 0$) that has no physical meaning which cannot be considered valid from the perspective of this work. However, for the different learning rates studied ($1e-2$, $1e-3$, $1e-4$, and $1e-5$), single generator only was capable of reconstructing the test trajectories for the represented learning rates and at the cost of an increase in the number of training epoch in case of $lr = 1e-4$ (from 6000 to 12000). Thus, in terms of stability in the solution, the robustness of the GENERIC formalism against the single generator must also be assessed, the latter being much more susceptible to failure.

Number of neurons per hidden layer

By increasing the number of neurons in hidden layers, both formalisms reduce the mean square error on the reconstructed system variables (Fig. 15). However, a high dependency on the network capacity has been observed for the single generator formalism. While the GENERIC formalism can reconstruct the trajectories even with 20 neurons in the hidden layer, the single generator is not able to reconstruct some trajectories with less than 50 neurons. If the learning rate is

decreased to 1e-4, the single generator reports an error in the reconstruction of the variables with less than 200 neurons in the hidden layer.

Appendix B. Influence of data in the learning process

Amount of data

The training database plays a key role in the learning process of the system. The quantity and diversity of the data are crucial to allow adequate learning with sufficient generalization. Thus, we analyzed the effect of increasing the database in both formalisms and for both examples (double thermo-elastic pendulum and Couette flow in an Oldroyd-B fluid).

By increasing the number of snapshots (decreasing the time step increment) a data augmentation is obtained. Moreover, the increase in the number of snapshots increases the number of integration points in the reconstruction of the system. Thus, even though the increase of snapshots will decrease the error of the net at each step of integration compared with the case of reference, the effect of the accumulated errors at each integration step will increase the error obtained (Fig. 11a, 12 a). However, in a detailed view of the learned system, the increase in the number of snapshots benefits the thermodynamic consistency of the learned system. Thus, the learned energy generators, the Hamiltonian, $\mathcal{H}$, and the Entropy, $\mathcal{S}$, in GENERIC, and, the free energy, $\mathcal{F}$ in the single generator, have been represented in Fig. 16. The energy conservation principle imposed in GENERIC implies that $\dot{\mathcal{H}} = 0$, which is better fulfilled for smaller time increments, as expected. As the number of snapshots increases, the results for the GENERIC formalism tend to converge to this imposed condition.

The double thermo-elastic is a chaotic system, thus increasing the number of trajectories in the database will introduce additional information on the solution space of the system. In the case of the Oldroyd-B fluid Couette flow, the increase in the trajectories is achieved by considering extra points in the discretization in the vertical direction of the model. In this sense, as the number of trajectories in the database increases, the error of the reconstruction of the vari-

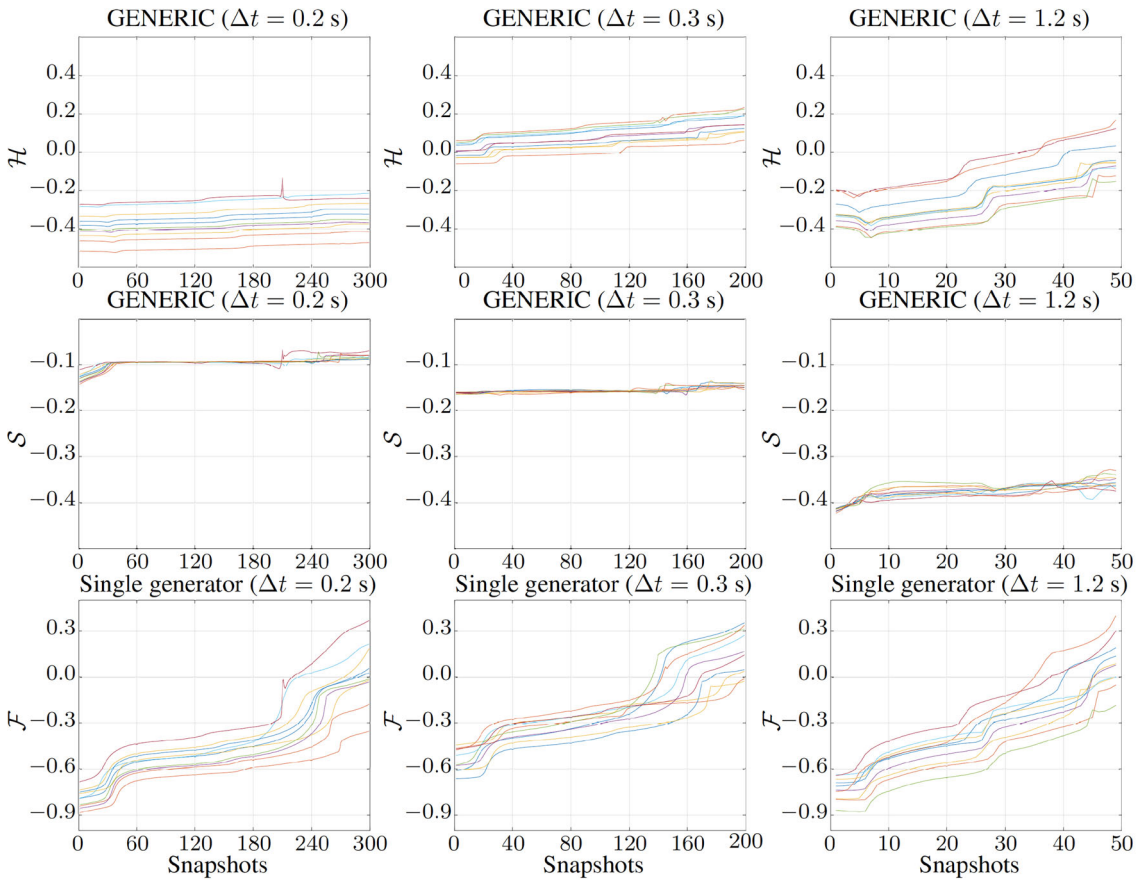

Fig. 16 Plot of the learned energy generators in GENERIC ($\mathcal{H}$, and $\mathcal{S}$) and single generator bracket ($\mathcal{F}$) for different runs of the double thermo-elastic pendulum. Increasing the number of snapshots improves thermodynamic consistency in GENERIC formalism

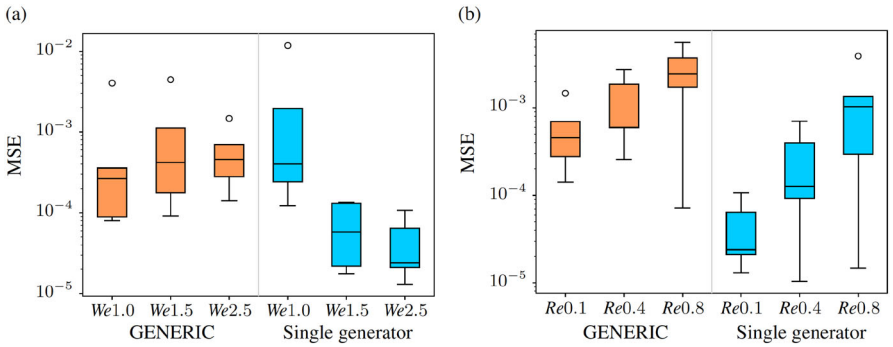

Fig. 17 Box plots for the data integration mean square error (MSE). **a** Errors corresponding to $Re = 0.1$ and increasing values for We number ($We = 1.0$, $We = 1.5$, and $We = 2.5$). **b** Errors corresponding to $We = 2.5$ and increasing values for Re number ($Re = 0.1$, $Re = 0.4$, and $Re = 0.8$)

ables of the system is reduced in both formalisms and for both analyzed systems (Fig. 11b, 12 b).

Influence of the physics

Different Reynolds, Re and Weissenberg, We numbers have been used in the generation of Couette flow databases in the Oldroyd-B fluid, to analyze the contributions of dissipative and conservative effects in the training of the two formalisms. As We increases, the elasticity of the system increases, which in turn decreases the relative importance of viscous effects and the dissipative effects. On the contrary, as Re is increased, the internal energy of the system increases which can be associated with the increase in the Hamiltonian. The effect of the variation of the We number is different in both models. Increasing We in GENERIC increases the error, while a reduction in the error is observed in the single generator formalism (Fig. 17a). However, for the minimum We value studied, GENERIC reports a lower error than the single generator. On the other hand, as Re is increased, the errors of both grow but not to the same extent, the effect on GENERIC being smaller, Fig. 17b. The tendency of the errors for both formalisms seems to imply that a higher increase in the Re number will eventually cause GENERIC to equal or improve the results of the single generator.

Whether due to the dissipative effect reduction or an increase in the conservative effect, the relationship between the conservative and dissipative effect seems to play a key role in the differences in the behavior of both formalisms. As the system becomes more dissipative the single generator seems to report better results. On the contrary, as the system becomes conservatively dominated (reduced $\mathcal{S}/\mathcal{H}$) the GENERIC reports better results. This can be justified by the separation of both energy generators and the degeneracy conditions in GENERIC, which gives support to learning the dissipative dynamics of the system even when their contribution is reduced. Besides, given that in single generator the energy generators are not separated, a reduction in dynamics can be diluted and lose relevance compared to the conservative part, increasing the error to a greater extent than in GENERIC in systems with a low dissipative component.

Acknowledgements This material is also based upon work supported in part by the Army Research Laboratory and the Army Research Office under contract/grant number W911NF2210271. This work was also supported by the Spanish Ministry of Science and Innovation, AEI/10.13039/501100011033, through Grants number PID2020-113463RB-C31 and TED2021-130105B-I00 and by the Ministry for Digital Transformation and the Civil Service, through the ENIA 2022 Chairs for the creation of university-industry chairs in AI, through Grant TSI-100930-2023-1. This research is also part of the DesCartes programme and is supported by the National Research Foundation, Prime Minister Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. The authors also acknowledge the support of ESI Group through the chairs at the University of Zaragoza and at ENSAM Institute of Technology.

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

References

1. Cuomo S, Cola VSD, Giampaolo F, Rozza G, Raissi M, Piccialli F (2022) Scientific machine learning through physics-informed neural networks: where we are and what's next. J Sci Comput 92(3):1–62
2. Cranmer M, Greydanus S, Hoyer S, Battaglia P, Spergel D, Ho S (2020) Lagrangian neural networks. arXiv Preprint arxiv:2003:04630
3. Mattheakis M, Sondak D, Dogra AS, Protopapas P (2022) Hamiltonian neural networks for solving equations of motion. Phys Rev E 105(6):065305
4. Raissi M, Perdikaris P, Karniadakis GE (2017) Physics informed deep learning (part i): data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711:10561
5. Mialon G, Garrido Q, Lawrence H, Rehman D, LeCun Y, Kiani BT (2023) Self-supervised learning with lie symmetries for partial

differential equations. Adv Neural Inf Process Syst 15(36):28973–9004
6. Pichi F, Moya B, Hesthaven JS (2024) A graph convolutional autoencoder approach to model order reduction for parametrized PDEs. J Comput Phys 501:1–30
7. Sosanya A, Greydanus S (2022) Dissipative hamiltonian neural networks: Learning dissipative and conservative dynamics separately. arXiv preprint: arXiv.2201: 10085
8. Cai S, Mao Z, Wang Z, Yin M, Karniadakis GE (2021) Physics-informed neural networks (pinns) for fluid mechanics: a review. Acta Mech Sin 37(12):1727–1738
9. Greydanus S, Dzamba M, Yosinski J (2019) Hamiltonian Neural Networks. Adv Neural Info Process Syst. 32
10. Hernandez Q, Badías A, González D, Chinesta F, Cueto E (2021) Deep learning of thermodynamics-aware reduced-order models from data. Comput Methods Appl Mech Eng 379:113763
11. Yu H, Tian X, Weinan E, Li Q (2021) OnsagerNet: learning stable and interpretable dynamics using a generalized Onsager principle. Phys Rev Fluids 6(11):114402
12. Gruber A, Lee K, Trask N (2023) Reversible and irreversible bracket-based dynamics for deep graph neural networks
13. Cueto E, Chinesta F (2023) Thermodynamics of learning physical phenomena. Arch Comput Methods Eng 30(8):4653–4666
14. Zhang Y, Lee J, Wainwright M, Jordan MI (2017) On the learnability of fully-connected neural networks. In: artificial intelligence and statistics, pp. 83–91, PMLR
15. Jou D, Casas-Vázquez J, Lebon G (1996) Extended irreversible thermodynamics. Reports on Progress in Physics 51(8):1105. https://doi.org/10.1088/0034-4885/51/8/002
16. Ma C, Wang J, et al. (2018) Model reduction with memory and the machine learning of dynamical systems. arXiv preprint arXiv:808:04258
17. González D, Chinesta F, Cueto E (2021) Learning non-markovian physics from data. J Comput Phys 428:109982
18. Kaufman AN (1984) Dissipative hamiltonian systems: a unifying principle. Phys Lett A 100(8):419–422
19. Grmela M (1984) Bracket formulation of dissipative fluid mechanics equations. Phys Lett A 102(8):355–358
20. Morrison PJ (1984) Bracket formulation for irreversible classical fields. Phys Lett A 100(8):423–427
21. Morrison PJ, Eliezer S (1986) Spontaneous symmetry breaking and neutral stability in the noncanonical hamiltonian formalism. Phys Rev A 33(4205):6
22. Edwards BJ (1998) An analysis of single and double generator thermodynamic formalisms for the macroscopic description of complex fluids. J Non-Equilib Thermodyn 23:301–333
23. Edwards BJ, Beris AN, Öttinger HC (1998) An analysis of single and double generator thermodynamic formalisms for complex fluids. ii. the microscopic description. J Non-Equilib Thermodyn 23:334–350
24. Beris AN (2001) Bracket formulation as a source for the development of dynamic equations in continuum mechanics. J Non-Newton Fluid Mech 96(1):119–136
25. Hernandez Q, Badias A, Chinesta F, Cueto E (2022) Thermodynamics-informed graph neural networks. IEEE Trans Artif Intell 5:1–1
26. González D, Chinesta F, Cueto E (2019) Thermodynamically consistent data-driven computational mechanics. Continuum Mech Thermodyn 31(1):239–253
27. Eldred C, Gay-Balmaz F (2020) Single and double generator bracket formulations of multicomponent fluids with irreversible processes. J Phys A: Math Theor 53(39):395701
28. Zaidni A, Morrison PJ, Benjelloun S (2024) Thermodynamically consistent cahn-hilliard-navier-stokes equations using the metriplectic dynamics formalism. arXiv preprint arXiv:2402:11116
29. Grmela M, Ottinger HC (1997) Dynamics and thermodynamics of complex fluids i development of a general formalism. Phys Rev E 56(6):6620
30. Beris AN, Edwards BJ (2024) Dissipation in nonequilibrium thermodynamics and its connection to the Rayleighian functional. Phys Fluids 36(1):13102
31. Hernández Q, Badías A, González D, Chinesta F, Cueto E (2021) Structure-preserving neural networks. J Comput Phys 426:109950
32. Morrison PJ, Eliezer S (1986) Spontaneous symmetry breaking and neutral stability in the noncanonical Hamiltonian formalism. Physical Review A 33(6):4205
33. Romero I (2009) Thermodynamically consistent time-stepping algorithms for non-linear thermomechanical systems. Int J Numer Meth Eng 79(6):706–732
34. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. Biochem Biophys Res Commun 498(1):254–261
35. Kingma DP, Adam JLB (2014) A method for stochastic optimization. In: international conference on learning representations, ICLR 2015—Conference track proceeding
36. Cherizol R, Sain M, Tjong J (2015) Review of non-newtonian mathematical models for rheological characteristics of viscoelastic composites. Green and Sustainable Chemistry 05(01):6–14
37. Binns J, Wynn A (2024) Global stability of Oldroyd-B fluids in plane Couette flow. J Nonnewton Fluid Mech 324:105171
38. Laso M, Öttinger HC (1993) Calculation of viscoelastic flow using molecular models: the connffessit approach. J Non-Newton Fluid Mech 47:1–20
39. Le Bris C, Leliévre T (2009) Multiscale modelling of complex fluids: a mathematical initiation. Lect Notes Comput Sci Eng 66 LNCSE:49–137