Rheologica Acta
https://doi.org/10.1007/s00397-022-01357-w

ORIGINAL CONTRIBUTION

Data-driven selection of constitutive models via rheology-informed neural networks (RhINNs)

Milad Saadat[1] · **Mohammadamin Mahmoudabadbozchelou**[1] · **Safa Jamali**[1]

Received: 2 February 2022 / Revised: 30 May 2022 / Accepted: 6 June 2022

Abstract

A myriad of empirical and phenomenological constitutive models that describe different observed rheologies of complex fluids have been developed over many decades. With each of these constitutive models' strength in recovering different rheological responses, algorithms that allow the data to automatically select the appropriate constitutive relations are of great interest to rheologists. Here, we present a rheology-informed neural network (RhINN) that enables robust model selection based on available experimental data with minimal user intervention. We train our RhINN on a series of experimental data for different complex fluids and show that it is capable of finding the appropriate model with the lowest number of fitting parameters for each data set. Finally, we show that uniform selection of a handful of data over the entire accessible shear rates does not affect the RhINN's accuracy, while providing a specific range of data (and omitting the rest) results in an erroneous model determination.

Keywords Rheology-informed neural network · Data-driven constitutive modeling · Physics-informed machine learning · Complex fluid meta-modeling

Introduction

The quest for modeling the rate-dependent stress response of complex fluids to an applied deformation is as old as rheology itself (Bingham 1916; Morrison 2001; Bird et al. 1987). The goal is to provide a closed-form mathematical expression describing the correspondence between the stress and deformation. Hence, with various types of rheological responses, many empirical and phenomenological constitutive models have also been developed (Soleymanzadeh et al. 2018; Fuchs and Ballauff 2005; Arora et al. 2017; Barthés-Biesel and Acrivos 1973). Of particular practical use are empirical models that describe the rate-dependence of shear viscosity as a simple scalar quantity, referred to as generalized Newtonian fluid (GNF) models. For instance, some of these classical GNF models that were developed to describe different polymeric liquids' steady-state flow curves (Bird 1965) include power law, Bingham, Herschel-Bulkley, and Carreau-Yasuda models (Morrison 2001; Herschel and Bulkley 1926). These models, although practically useful, are strictly limited to the steady-state response of common rate-dependent systems and cannot describe the memory or elastic effects or multi-component systems that show more complex behaviors. Thus, models with added levels of complexity have been developed over time to recover the rich rheology of different materials. For instance, a series of thixotropic elasto-visco-plastic (TEVP) models have been developed to describe the long-time transients, static and dynamic yield stresses measured, and hysteretic effects in time-dependent materials with a fading memory of deformation history (Dimitriou and McKinley 2014; De Souza Mendes 2011; Larson 2015; Larson and Wei 2019; Armstrong et al. 2016; Joshi 2022). When solved for the quasi-steady-state response, these models also recover a non-monotonic flow curve (over a limited range of shear rates) with the shear stress decreasing as a function of shear rate, and two (static and dynamic) yield stress values that cannot be recovered using GNFs.

Regardless of the type of material, the common practice in describing the material rheology is to interrogate its response through rheometry and then seek models that can best simulate the observed behavior. By simply looking at a material's steady-state stress (or viscosity) versus shear rate

---

✉ Safa Jamali
s.jamali@northeastern.edu

[1] Department of Mechanical and Industrial Engineering, Northeastern University, Boston 02115, MA, USA

curve, a seasoned rheologist can discard a handful of models irrelevant to that particular sample. However, even the steady-state flow curve of the shear viscosity versus applied deformation rate can be complex in most complex fluids, making the model selection a non-trivial task. Thus, frameworks that aid in model selection can be of great practical use. For instance, different TEVP models described above can be used to model different types of behavior observed in thixotropic materials. Nonetheless, the number of necessary parameters to recover a specific behavior can significantly differ from one model to another, resulting in an entirely different set of experiments required for parameterizing the model. Hence, time- and cost-effective tools that can automatically differentiate between the models and their resulting rheologies and identify the one with the least number of parameters without loss of accuracy are of great interest.

Machine learning (ML) algorithms can make a transformative leap in this area. These frameworks can explore massive design spaces (Karniadakis et al. 2021), handle real-time tasks (Ritto and Rochinha 2021), identify multi-dimensional correlations (Sun and Barnard 2019; Sun et al. 2013), and manage ill-posed problems (Gao et al. 2022). ML frameworks can be categorized into supervised, semi-supervised, and unsupervised learning (Brunton et al. 2020). Neural networks (NNs) as means to correlating the complex relations between the inputs and outputs by forming a data-driven framework have been widely used in semi-supervised (e.g., deep reinforcement learning (Arulkumaran et al. 2017)) and unsupervised (e.g., dimensionality reduction through self-organizing maps (Kohonen 2001)) tasks. However, neural networks are most recognized (and used) in supervised learning problems. While purely data-driven NN models may fit observations very well, the prediction step might become unrealistic and result in reduction of the overall generalization performance (Karniadakis et al. 2021).

Therefore, physics-informed learning is needed, where the physical domain knowledge is included in every training step (Wang et al. 2017) to improve the performance of a learning algorithm. Physics-informed neural networks (PINNs), as a recent example of this learning methodology, have shown remarkable performance by leveraging the input data using governing physics equations to further augment the prediction accuracy (Raissi et al. 2019; Raissi et al. 2020; Penwarden et al. 2022; Zhu et al. 2021; Cai et al. 2022; Cuomo et al. 2022). Recently, we have reported a number of rheology-informed neural networks (RhINNs) that share roots with PINNs but adhere to rheological intuitions in their architecture (Mahmoudabadbozchelou and Jamali 2021; Mahmoudabadbozchelou et al. 2021; Mahmoudabadbozchelou et al. 2022).

Generally, in RhINNs, predictions can be made in two ways: first, we may use the framework in a *forward* direction with known boundary and initial conditions as input data to solve a constitutive model in a spatiotemporal domain. On the other hand, one might need to interpret data gathered from a rheometer (or numerical analysis) to recover the hidden rheology of a sample. In this case, only a handful of data will be used in an *inverse* platform along with a constitutive model with yet-to-be-determined parameters. The ultimate goal here is to unravel the hidden rheology of a sample by confining those fitting parameters.

Here we report on a RhINN platform capable of choosing the most accurate and efficient constitutive model (describing a particular rheological behavior) from a set of input data. In particular, we asked the following question: Can a handful of scattered data from a sample help us identify and choose the best model representing that specific sample with minimal user intervention? The goal is to develop a framework in which (*i*) the data automatically finds the best model with minimal user input (such as priors and bounds), and (*ii*) model parameters are accurately recovered. In other words, we seek to provide a platform that can include any rheological constitutive equation of interest, regardless of their complexity (e.g., tensorial forms of fully resolved non-linear viscoelastic models) and recover the material constants via a completely automated procedure. Thus, the algorithm must automatically test several constitutive equations against the given ground-truth/provided data. However, as a proof of concept, the solution of multiple GNF models, as well as more complex thixotropic visco-plastic (TVP) models for quasi-steady-state flow curve, is used here for the development of the platform. Upon establishing the performance of this ML platform, we systematically remove the included data to interrogate the efficiency and accuracy of the algorithm in finding the appropriate model.

Problem setup and methodology

Constitutive models

As previously described, several constitutive equations can be chosen to represent a complex fluid's steady-state shear stress response. Here, we select nine (9) different models with a different number of parameters to be embedded into our neural network. The goal is to provide a relationship between the shear stress, σ (or viscosity, η), and the imposed shear rate, $\dot{\gamma}$. These nine models, along with their shear stress/viscosity form and parameters, are summarized in Table 1.

Rheology-informed neural network

The NN interacts with the constitutive models in two iterative steps: first, all models' initial (or previous) parameters are fed into the NN, and a loss function is calculated (the

Table 1 Different constitutive models implemented into the neural network to determine the hidden rheology of different samples

Model	Shear stress/viscosity form	Parameters
Power law (PL) (Morrison 2001)	$\sigma = K\dot{\gamma}^n$	K, n
Herschel-Bulkley (HB) (Herschel and Bulkley 1926)	$\sigma = \sigma_y + K\dot{\gamma}^n$	σ_y, K, n
Bingham (BH) (Morrison 2001)	$\eta = \begin{cases} \infty & if \sigma \leq \sigma_y \\ \eta_0 + \frac{\sigma_y}{\dot{\gamma}} & if \sigma > \sigma_y \end{cases}$	η_0, σ_y
Carreau-Yasuda (CY) (Morrison 2001)	$\eta = \eta_\infty + \left(\eta_0 - \eta_\infty\right)\left[1 + (\dot{\gamma}\lambda)^a\right]^{\frac{n-1}{a}}$	$\eta_\infty, \eta_0, \lambda, a, n$
Three-component (TC) (Caggioni et al. 2020)	$\sigma = \sigma_y + \sigma_y\left(\frac{\dot{\gamma}}{\dot{\gamma}_c}\right)^{\frac{1}{2}} + \eta_{bg}\dot{\gamma}$	$\sigma_y, \dot{\gamma}_c, \eta_{bg}$
TCC (Mahmoudabadbozchelou et al. 2021)	$\sigma = \sigma_y + \sigma_y\left(\frac{\dot{\gamma}}{\dot{\gamma}_{cTC}}\right)^{\frac{1}{2}} + \dot{\gamma}k\left[1 + \left(\frac{\dot{\gamma}}{\dot{\gamma}_c}\right)^2\right]^{-0.5}$	$\sigma_y, \dot{\gamma}_{cTC}, k, \dot{\gamma}_c$
Casson (Macosko 1994)	$\sqrt{\sigma} = \sqrt{\sigma_y} + \sqrt{\eta_0\dot{\gamma}}$	σ_y, η_0
TVP (Mahmoudabadbozchelou and Jamali 2021)	$\sigma = \sigma_y\lambda_0 + \eta_s\dot{\gamma} + \eta_p\lambda_0\dot{\gamma},\ \lambda_0 = \frac{k^+}{k^+ + k^-\dot{\gamma}}$	$\sigma_y, k^+, k^-, \eta_s, \eta_p$
Steady-state IKH (Dimitriou and McKinley 2014)	$\sigma = \eta_p\dot{\gamma} + \frac{c}{q} + \frac{k_1k_3}{k_1 + k_2\dot{\gamma}}$	$\eta_p, \frac{c}{q}, k_1, k_3, k_2$

These nine models have between two and five fitting parameters each, which will be trained along with the NN variables and biases to reduce the total loss. *TCC*, *TVP*, and *IKH* stand for three-component Carreau, thixotropic visco-plastic, and isotropic kinematic hardening, respectively

upward green arrow in Fig. 1). Second, through a feed-forward back-propagation algorithm, the NN will correct its parameters along with all the fitting parameters enumerated in Table 1 to minimize the loss function (the downward blue arrow in Fig. 1).

The schematic illustration of the RhINN architecture is detailed in Fig. 2. Since the only output of interest here is the (quasi) steady-state solutions of simple constitutive models, the only meaningful input to the NN would be the shear rate, $\dot{\gamma}$. However, the ultimate goal is to develop a unified toolbox consisting of *all* rheological models under one umbrella. Therefore, a model[1] was built by sub-classing a `TensorFlow Keras` model (Blechschmidt and Ernst 2021) and implementing a functional application programming interface (API), in which a unique loss function is calculated for each rheological model. The total loss function (ϕ^i) for each constitutive model (i) is defined as:

$$\phi^i = \phi^i_d + \phi^i_f \tag{1}$$

where:

$$\phi^i_d = \mathrm{MSE}\left(\sigma^i_p, \sigma^i_{gt}\right) = \frac{1}{n}\sum_{k=1}^{n}\left(\sigma^i_{p,k} - \sigma^i_{gt,k}\right)^2 \tag{2}$$

is the loss corresponding to the discrepancies between the ground-truth (σ^i_{gt}) and the predicted shear stress, σ^i_p for each constitutive model (which is based on the mean squared error, MSE), n is the total number of given data ($\sigma - \dot{\gamma}$ data points for each sample), and:

$$\phi^i_f = \mathrm{Res}^i = \frac{1}{n_{fp}}\sum_{p=1}^{n_{fp}}\left(\mathrm{Res}^i_p\right)^2 \tag{3}$$

is the residual (Res) calculated from each constitutive model, and n_{fp} is the number of artificial function points (shear rate points) that we use to calculate the residual. For instance, and for the case of the power-law model, ϕ^i_f in 3 would be $\frac{1}{n_{fp}}\sum_{p=1}^{n_{fp}}\left(\sigma_p - K\dot{\gamma_p}^n\right)^2$, where K and n are calculated by the NN, $\dot{\gamma}_p$ is the shear rate based on the artificial points, and σ_p is the NN prediction of the shear stress. After the loss calculation, through back-propagation, the NN variables, biases, and model parameters are corrected to reduce and eventually minimize the total loss.

The usual issue of fitting bias at high shear stress levels (in contrast to lower ones) was resolved by scaling the shear stress in our $\sigma - \dot{\gamma}$ data, training the model and then re-scaling the shear stress back into its original value after the training step. Thus, logarithmic shear stress values were

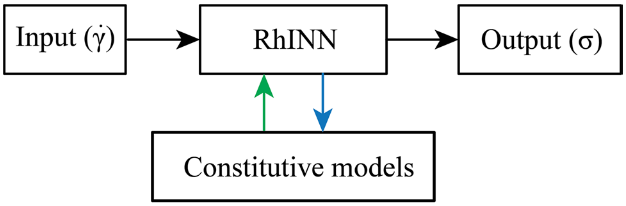

Fig. 1 The overall model-selector RhINN architecture implemented in this work. Embedded constitutive models interact with the RhINN platform in two ways. Model parameters are used to calculate losses, followed by optimization of the NN variables and biases, along with model parameters, to minimize the calculated loss

---

[1] https://github.com/procf/RhINNs

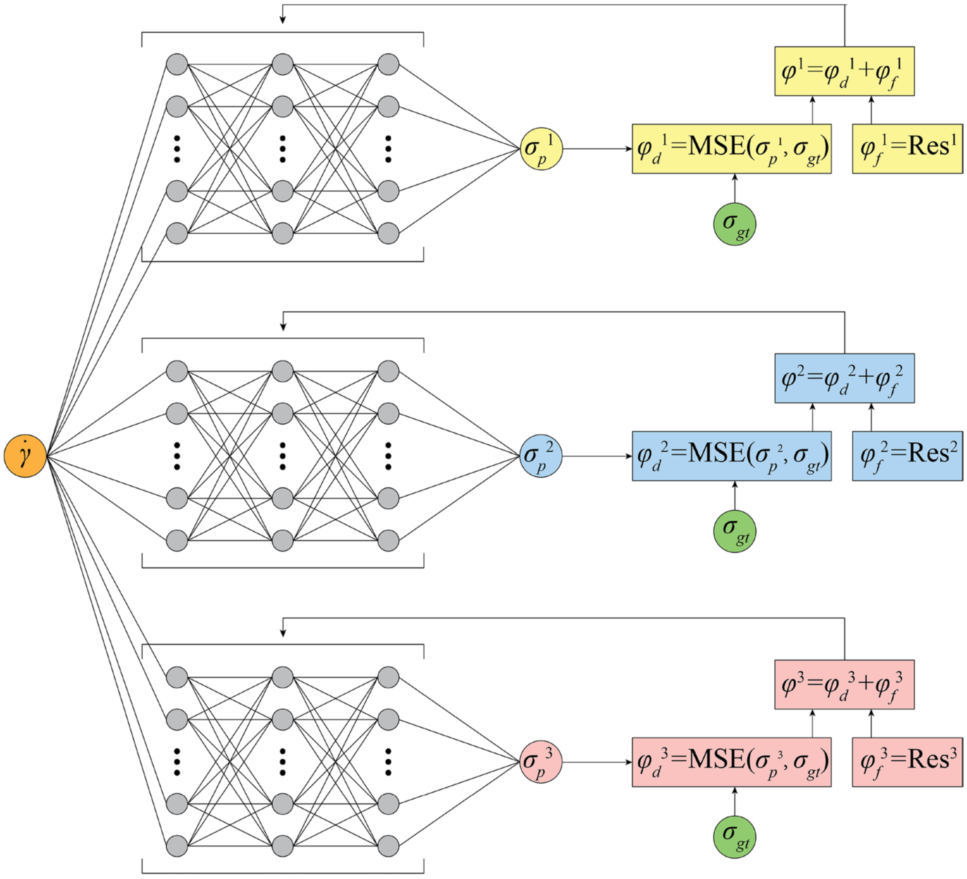

Fig. 2 Schematic illustration of the RhINN architecture. Here, only three constitutive models and three hidden layers for each model are shown for aesthetic purposes. Each constitutive model has a corresponding data discrepancy (ϕ_d) and function residual (ϕ_f). This fact is also stressed by color-coding the stress prediction and the loss function for each constitutive model. The ground-truth shear stress, σ_{gt}, is the same for all constitutive models. The summation of ϕ_d and ϕ_f for each constitutive model is fed into the NN, and NN variables and biases are adjusted to minimize the total error

used (instead of their absolute values) to mitigate the bias at high shear stresses of the flow curve. This way, the model accuracy in low shear rates is also preserved.

For each sample, all nine constitutive models are called, and the Adam optimizer is allowed to perform 60,000 iterations with a learning rate of 1×10^{-4}, which yields ~ 306s of total calculation on average to finish on a MacBook Pro (M1 Max, 64 GB RAM). The program is based on `TensorFlow 2.7.0`, and sample swap occurs with minimal user intervention. The number of neurons per layer and the number of layers used within an NN are commonly referred to as its hyperparameters. The role of these hyperparameters was investigated by changing the number of hidden layers from 1 to 16 (1, 2, 4, and 16) and by setting the number of neurons to 10, 25, and 50. For all values of layer and neuron count, negligible variation was seen in the total error for each constitutive equation (ϕ^i). Thus, to find a suitable compromise between the computational cost and accuracy, four hidden layers, with 25 neurons (nodes) in each layer, were chosen for our RhINN. A `tanh` activation function is employed for all hidden layers and all models. A `ReLU` activation function was also tested, but `tanh` activation function exhibited better overall stability due to the data normalization mentioned above. NN variables and biases are initialized using the `glorot_normal` method. The nine models summarized in 1 have 31 fitting parameters total, all of which are initialized to unity with a small noise to prevent instability in the first few iterations. The parameters are allowed to vary without any constraint, thus keeping the platform generalizable to any other rheological data set. The unconstrained nature of our algorithm has important consequences, as will be explained in sections "RhINN: convergence and benchmark"-"Model selection and recovered parameters."

Experimental data

In order to test and validate our proposed architecture, we used experimental data from the literature for a variety of materials and rheological characterizations. These samples, along with their descriptions, are summarized in Table 2. Here, we used data of solutions of partially hydrolyzed polyacrylamide with 1600×10^4 molecular weight (HPAM) at two different concentrations of 1000 mg L^{-1} and 2000 mg L^{-1} (Huang et al. 2019), suspension of $\Phi = 3\%$ carbon black dispersed in a mineral oil (Dagès et al. 2021), a heavy mineral oil with a paraffin wax at two different concentrations of 5% and 10% (Dimitriou and McKinley 2014), a complex colloid/worm-like micellar (WLM) mixture at three different concentrations of salt in the background fluid (Mahmoudabadbozchelou et al. 2021), colloidal gelatin particles in distilled water with two volume fractions of 0.2 and 0.05, (Nair et al. 2019), a worm-like micellar solution of 100 mM CTAB and

Table 2 Experimental samples used in this work to test our RhINN platform

No.	Type	Description
1	Polymer solution	Partially hydrolyzed polyacrylamide (HPAM), 1000 mg L^{-1} (Huang et al. 2019)
2	Polymer solution	Partially hydrolyzed polyacrylamide (HPAM), 2000 mg L^{-1} (Huang et al. 2019)
3	Carbon black gel	3 % carbon black in a mineral oil (Dagès et al. 2021)
4	Waxy oil	Heavy mineral oil with a 5% paraffin wax (Dimitriou and McKinley 2014)
5	Waxy oil	Heavy mineral oil with a 10% paraffin wax (Dimitriou and McKinley 2014)
6	Colloidal and WLM mixture	A complex colloid/WLM mixture, salt level 1 (Mahmoudabadbozchelou et al. 2021)
7	Colloidal and WLM mixture	A complex colloid/WLM mixture, salt level 2 (Mahmoudabadbozchelou et al. 2021)
8	Colloidal and WLM mixture	A complex colloid/WLM mixture, salt level 3 (Mahmoudabadbozchelou et al. 2021)
9	Colloidal gel	Colloidal gelatin particles in distilled water, volume fraction (Φ) = 0.2 (Nair et al. 2019)
10	Colloidal gel	Colloidal gelatin particles in distilled water, $\Phi = 0.05$ (Nair et al. 2019)
11	Worm-like micellar solution	A worm-like micellar solution of A NaSal surfactants in DI water (Cardiel et al. 2013)
12	Polymer melt	Low-density polyethylene melt at 423 K (Dunstan 2019)
13	Polymer melt	Low-density polyethylene melt at 463 K (Dunstan 2019)

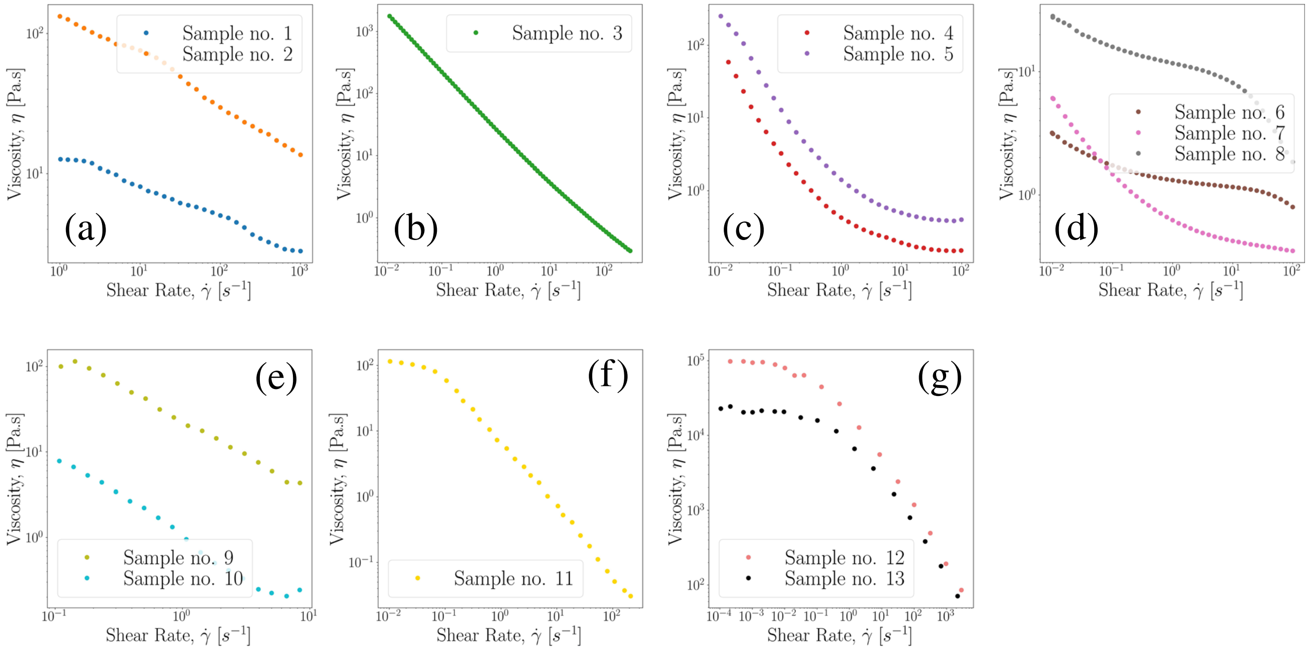

Fig. 3 Collection of steady-state experimental data digitized and re-plotted from the literature: **a** solutions of partially hydrolyzed polyacrylamide with 1600×10^4 molecular weight (HPAM) at two different concentrations of 1000 mg L^{-1} and 2000 mg L^{-1} (Huang et al. 2019), **b** suspension of $\Phi = 3\%$ carbon black dispersed in a mineral oil (Dagès et al. 2021), **c** a heavy mineral oil with a paraffin wax at two different concentrations of 5% and 10% (Dimitriou and McKinley 2014), **d** a complex colloid/worm-like micellar mixture at three different concentrations of salt in the background fluid (Mahmoudabadbozchelou et al. 2021), **e** colloidal gelatin particles in distilled water with two volume fractions (Φ) of 0.2 and 0.05 (Nair et al. 2019), **f** a worm-like micellar solution of 100 mM CTAB and 32 mM NaSal surfactants in DI water (Cardiel et al. 2013), and **g** low-density polyethylene melts at two temperatures of $423K$ and $463K$ (Dunstan 2019)

32 mM NaSal surfactants in DI water (Cardiel et al. 2013), and low-density polyethylene melts at two temperatures of 423 K and 463 K (Dunstan 2019). These samples are referred to with their assigned numbers in 2 hereafter.

All data collected are plotted in a series of flow curves in Fig. 3, indicating a total range of over six decades for the variation of the viscosity and shear rate. Since one has to ensure that the NN retains a robust model recovery irrespective of the range of input parameters or stresses, the range of collected data here presents a suitable benchmark for our developed RhINN platform.

Once the RhiNN starts the course of finding the model parameters and minimizing the loss functions accordingly, the choice between two (or more) competing constitutive models can become tricky or misleading. This is because for some materials, multiple models, especially with an increasing number of parameters, can recover the same set of behavior. Therefore, a reliable figure of merit for model selection is essential in ensuring that models are appropriately chosen. To this end, we used the total error for each model, ϕ^i divided by the maximum shear stress of that particular sample (σ^i_{max}), as the selection metric, meaning that the model with the lowest $\phi^i/\sigma^i_{\text{max}}$ (in Pa^{-1}) can best describe the rheological phenomenon. Moreover, if two (or more) models have $\phi^i/\sigma^i_{\text{max}}$ smaller than $5 \times^{-4}$ Pa^{-1}, the model with a smaller number of fitting parameters is selected. This is to ensure that a given rheological measurement is described through the simplest and most accurate constitutive model without resorting to unnecessary complexities within the model itself. Although other model selection heuristics, such as Bayesian inference criterion (Freund and Ewoldt 2015) or adaptive parallel tempering (Armstrong et al. 2017), have been successfully employed, we used the MSE norm (2, 3), as widely selected in physics-informed learning.

Results and discussion

RhINN: convergence and benchmark

For any NN problem, it is vital to assess whether the NN can successfully minimize the losses. To this end, $\phi^i/\sigma^i_{\text{max}}$ as a function of the iteration number is plotted in Fig. 4 for sample no. 2 (see Table 2). As seen in this figure, $\phi^i/\sigma^i_{\text{max}}$ (or the total error, ϕ^i) plateaus after ≈ 35,000 iterations for sample no. 2. However, the program is allowed to iterate for another 25,000 iterations to ensure that all errors for different models have reached an acceptable steady-state accuracy. While loss functions are presented for this collected sample, similar trends are observed for all different materials/samples studied throughout this work.

Another crucial step is to see where RhINN stays compared to other numerical and probabilistic alternatives. Thus, we benchmarked our algorithm with two other methods, i.e., trust region reflective, TRF (as implemented in `SciPy v1.7.3`, a `Python` package), and a Bayesian Inference Criterion (BIC) method (using `PyMC3 v3.11.5`). We also used the current RhINN platform *without* adding the residuals from the constitutive models to see if including the physics is advantageous or not. In other words, we set 3 equal to zero, which converts RhINN to a purely statistical neural network, i.e., deep neural network, DNN. We benchmarked our method for both accuracy (in terms of root mean square (RMS) of prediction errors from the experimental data over the maximum shear stress, σ_{max}) and the computing time (parameter recovery runtime in s), and the results are presented in Table 3. As can be seen in this table, RhINN's prediction accuracy is comparable with the other two methods and is also comparable in the runtime with BIC. However, both RhINN and BIC are outperformed by TRF in runtime. Moreover,

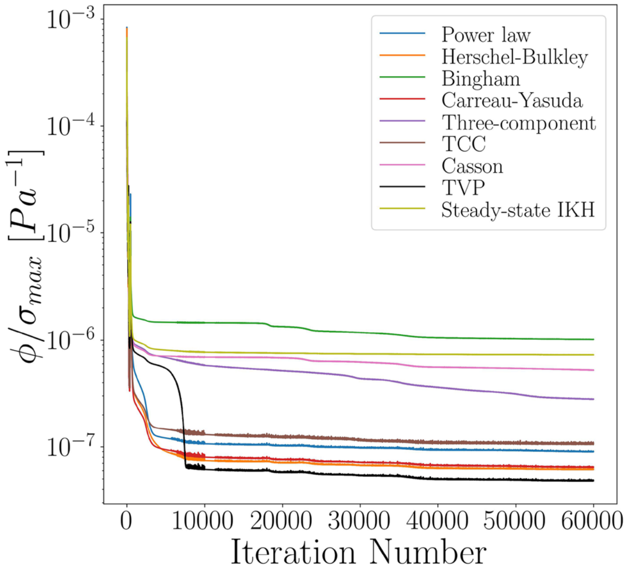

Fig. 4 Total error divided by the maximum shear stress ($\phi^i/\sigma^i_{\text{max}}$) history as a function of the iteration number for the 2000 mgL^{-1} polymer solution sample (no. 2) for all constitutive models. After ≈ 35,000 iterations, $\phi^i/\sigma^i_{\text{max}}$ for this sample (as well as other samples) plateaus. However, the program is allowed to run for 60,000 iterations to ensure accuracy

Table 3 Error (in terms of root mean square (RMS) of prediction errors from the experimental data over the maximum shear stress,σ_{max}) and runtime (in s) comparison of our developed model (RhINN) with a trust region reflective (TRF) method and a Bayesian Inference Criterion (BIC) method

Sample	Method	RMS/σ_{max}(Pa^{-1})	Runtime (s)
2	RhINN	0.144	14.75
	TRF	0.143	0.56
	BIC	0.145	14.42
	DNN	28.441	11.18
8	RhINN	0.456	14.34
	TRF	0.455	0.17
	BIC	0.459	24.7
	DNN	76.661	11.30
10	RhINN	2.166	13.75
	TRF	2.157	0.08
	BIC	2.240	14.61
	DNN	2.948	10.95

The same RhINN architecture without imposing the constitutive models' losses (3) is also included as a reference and is denoted with deep neural network, *DNN*. All three models are asked to recover parameters of the TCC model

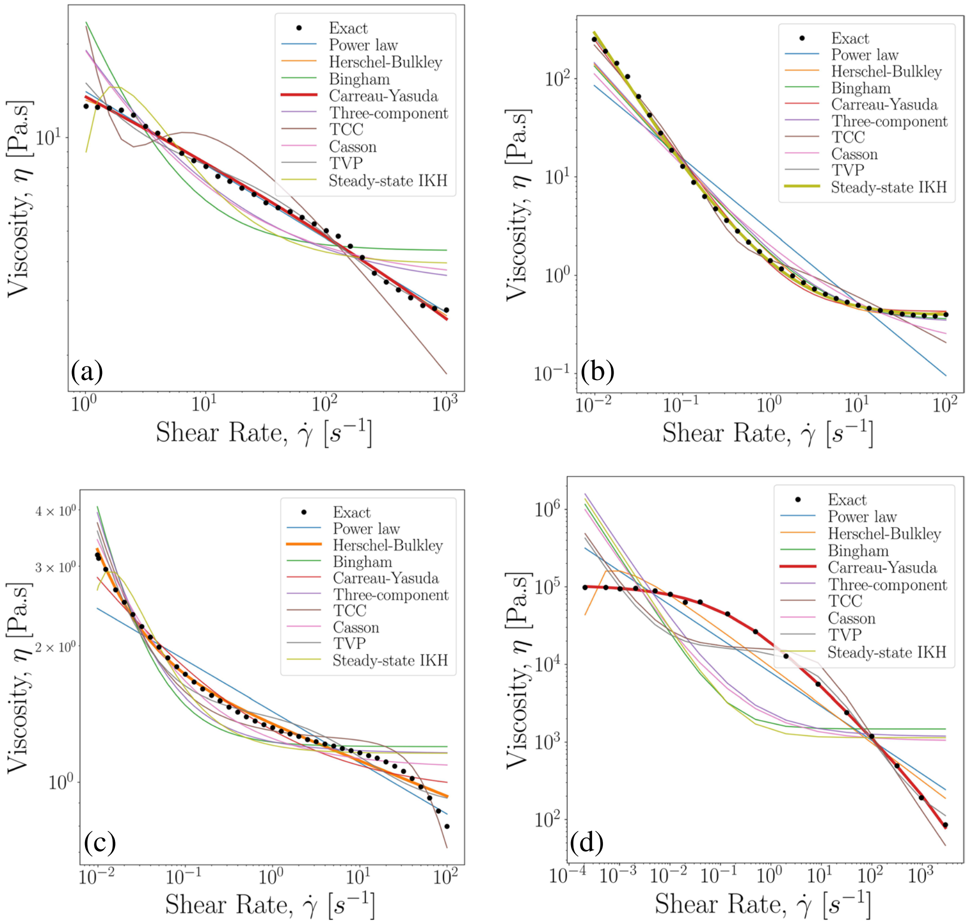

Fig. 5 Viscosity behavior as a function of the shear rate for four experimental samples after 60,000 iterations. **a** Polymer solution with a 1000 mg L^{-1} concentration (sample no. 1); **b** 10% waxy oil (sample no. 5), **c** a colloidal and WLM mixture, salt level 1 (sample no. 6); and **d** polymer melt at 423 K (sample no. 12). The best model for each sample is shown with full opacity and a thicker line, with predictions of all other models presented in the background for visual comparison

relinquishing the proper physics is quite detrimental to the model accuracy, as seen in the DNN cases. The critical point here is that both TRF and BIC strictly rely on the range of initial conditions (or priors for BIC) and predetermined constraints to converge. In other words, TRF and BIC fail to converge when the four parameters of TCC are left unbounded. However, RhINN can run and converge without constraints on the parameter range. If more precision is needed, the predicted parameters by RhINN can be fed into either TRF or BIC methods as initial conditions (or priors) to improve the accuracy.

RhINNs: steady-state flow curve

The developed platform for parameter identification was tested against the experimental data depicted in Fig. 3. Here, as an example, the steady-state viscosity vs. shear rate response for sample nos. 1, 5, 6, and 12 is shown in Fig. 5. It is worth mentioning here that these shear stress predictions are generated by inserting the recovered parameters (Table 5) from our RhINN platform into the constitutive models listed in Table 1. Thus, these represent the best fits that the NN predicts/recovers for any given data set. Notably, we had shown previously that a combination of experimental data as high fidelity input, and synthetically generated data from constitutive models as the low fidelity input, can result in accurate predictions of the flow curve for different complex fluids (Mahmoudabadbozchelou et al. 2021). Nonetheless, here RhINNs are used to find the best model from input data and not to make predictions of the flow curve. In Fig. 5, (a) for the polymer solution (sample no. 1) the Carreau-Yasuda model, (b) for the waxy oil (sample no. 5) the IKH model, (c) for the colloid and WLM mixture (sample no. 6) the Herschel-Bulkley model, and (d) for the polymer melt (sample no. 12) the Carreau-Yasuda model are found to best describe the flow curves. However, as briefly explained in the “Introduction” section, the choice can become tricky when two models have an indistinguishable agreement with the exact solution, making the selection prone to error. Thus, our selected figure of merit, i.e., $\phi^i/\sigma^i_{\max}$, along with the number of fitting parameters, is employed to facilitate the model selection.

Table 4 Total error divided by the maximum shear stress ($\phi^i/\sigma^i_{\max}$, in Pa^{-1}) after 60,000 iterations for samples shown in Fig. 3 and for models formulated in Table 1

Sample	PL (2)	HB (3)	BH (2)	CY (5)	TC (3)	TCC (4)	Casson (2)	TVP (5)	IKH (5)
1	2.50×10^{-7}	2.35×10^{-7}	3.33×10^{-6}	2.30×10^{-7}	1.50×10^{-6}	2.33×10^{-6}	1.68×10^{-6}	3.09×10^{-7}	1.90×10^{-6}
2	9.55×10^{-8}	6.43×10^{-8}	1.04×10^{-6}	7.56×10^{-8}	4.06×10^{-7}	1.15×10^{-7}	5.74×10^{-7}	5.12×10^{-8}	7.36×10^{-7}
3	2.49×10^{-5}	1.60×10^{-6}	5.33×10^{-5}	9.55×10^{-6}	2.67×10^{-6}	1.76×10^{-6}	1.22×10^{-5}	4.19×10^{-6}	4.19×10^{-6}
4	2.78×10^{-3}	5.23×10^{-4}	5.37×10^{-4}	2.44×10^{-4}	5.43×10^{-4}	6.07×10^{-4}	1.12×10^{-3}	1.60×10^{-4}	1.12×10^{-4}
5	1.39×10^{-3}	1.85×10^{-4}	1.88×10^{-4}	7.95×10^{-5}	2.78×10^{-4}	2.12×10^{-4}	4.15×10^{-4}	4.85×10^{-5}	4.17×10^{-5}
6	1.60×10^{-5}	3.73×10^{-6}	3.50×10^{-5}	7.52×10^{-6}	2.60×10^{-5}	9.00×10^{-6}	1.21×10^{-5}	6.52×10^{-6}	2.34×10^{-5}
7	3.11×10^{-4}	1.77×10^{-5}	1.23×10^{-4}	1.44×10^{-5}	3.71×10^{-5}	2.85×10^{-5}	8.52×10^{-6}	1.13×10^{-5}	6.41×10^{-5}
8	4.26×10^{-5}	3.98×10^{-5}	1.74×10^{-4}	1.59×10^{-5}	1.53×10^{-4}	1.33×10^{-5}	1.23×10^{-4}	1.02×10^{-5}	1.49×10^{-4}
9	1.36×10^{-5}	1.37×10^{-5}	3.80×10^{-5}	1.45×10^{-5}	1.34×10^{-5}	1.46×10^{-5}	2.35×10^{-5}	1.37×10^{-5}	1.42×10^{-5}
10	8.19×10^{-4}	4.46×10^{-4}	6.78×10^{-4}	2.33×10^{-4}	7.28×10^{-4}	2.41×10^{-4}	7.60×10^{-4}	4.69×10^{-4}	4.69×10^{-4}
11	1.44×10^{-3}	1.33×10^{-3}	1.64×10^{-3}	4.73×10^{-4}	1.40×10^{-3}	9.74×10^{-4}	1.20×10^{-3}	1.22×10^{-3}	1.15×10^{-3}
12	2.49×10^{-7}	1.63×10^{-7}	1.02×10^{-6}	8.00×10^{-8}	8.74×10^{-7}	3.24×10^{-7}	9.17×10^{-7}	2.46×10^{-7}	1.17×10^{-6}
13	3.14×10^{-7}	2.31×10^{-7}	1.18×10^{-6}	5.17×10^{-8}	1.23×10^{-6}	3.38×10^{-7}	1.17×10^{-6}	2.89×10^{-7}	1.38×10^{-6}

Smaller values correspond to a better agreement with the experimental (σ_{gt}) data, and the best model for each sample is highlighted in red. The numbers next to the model names are the models' parameter count

Model selection and recovered parameters

From Table 4, it can be inferred that even for the samples with the simplest viscosity behavior, two-parameter models, e.g., power law, Bingham, and Casson, to a large extent, fail to reduce the total error. Interestingly, the more complex models do not necessarily attain lower error values. For instance, the TCC model was developed explicitly for sample nos. 6, 7, and 8. However, TCC does not yield the lowest error for sample no. 6 (as shown in Fig. 5); instead, the Herschel-Bulkley model has the lowest loss function. This is because for the salt level presented in sample no. 6, the second plateau observed for the system vanishes. Thus, a simple yield stress constitutive model can describe the behavior without the need for four different model parameters in the TCC equation. For samples with a unique shear-thinning profile, e.g., the polymer melt (sample nos. 12 and 13) studied here, having more fitting parameters is indeed helpful in finding the best model. That is why the Carreau-Yasuda model, with five fitting parameters, can outperform other constitutive models.

The entirety of recovered parameters from our RhINN platform for different samples studied is summarized in Table 5. In this routine, no constraint was imposed during the training steps. Thus, we can logically anticipate that some of these constitutive models have non-physical fitting parameters. For instance, the Herschel-Bulkley model predicts σ_y=-1.008Pa for the worm-like micellar solution (sample no. 11), which is unrealistic; however, for all the selected models shown in red, all model parameters are within a reasonable range.

Model selection for smaller data sets

From a practical perspective, obtaining the flow curves presented in Fig. 5 needs individual stress growth tests at varying shear rates, which can be time-consuming. Thus, methods that can find the appropriate model (that describes the observed rheology) with minimal experimental measurements can be beneficial. Here, to interrogate the applicability of RhINN model selection with a smaller number of data provided, we systematically reduced the size of available data for a particular sample with different shear-thinning profiles, i.e., sample no. 8, and reported on the model selection. In particular, we seek to answer the following question: Between the *range* and *size* of data, which one should be prioritized? To answer this question, and in the first step, the given data were split into three regions of low (0.01 to 0.16s^{-1}), moderate (0.2 to 4s), and high (5 to 100s^{-1}) shear rates and were used to train the RhINN with the parameters mentioned in the "Rheology-informed neural network" section, and the results are shown in Fig. 6. These results suggest that by sampling the given data in a specific region of shear rates applied, only some of the constitutive models can capture the flow curve within that specific data range; however, even for those models determined, the parameters do not necessarily remain valid when going to another range of shear rates. For instance, in both the intermediate and high shear rate regimes (Fig. 6b and c), the TVP model results in the best description of the observed data. Nonetheless, by looking at the TVP curves in Fig. 6b and c, it is clear that the model parameters are significantly different for these two separate ranges of data provided. The role that the range of provided data can play is even more apparent in Fig. 6a, where the Carreau-Yasuda model is recovered for the low shear rate regime. Note that for the same sample and providing the entire range of data at hand, the TVP model was found to be the best descriptor of the flow curve (see Table 4). As such, for all three shear rate regimes shown in Fig. 6, the RhINN platform recovers a somewhat inaccurate set of model parameters, although, for two regimes, the model selection remains valid. This is expected, as the flow curve for sample no. 8 shows a distinct set of behaviors at these different regimes. As such, for a sample that does not show

Table 5 Recovered parameters for the set of constitutive equations presented in Table 1 implemented in this work and for experimental samples (columns) shown in Fig. 3

Model	Params	1	2	3	4	5	6	7	8	9	10	11	12	13
PL	η_0 (Pa • s)	7.45×10^{-2}	1.11×10^{-1}	7.25×10^{-1}	3.79×10^{-1}	4.25×10^{-1}	9.07×10^{-1}	3.79×10^{-1}	1.47	1.16	7.84×10^{-1}	1.47	3.5	7.84×10^{-1}
	n	7.63×10^{-1}	6.57×10^{-1}	1.34×10^{-1}	4.24×10^{-1}	2.62×10^{-1}	8.87×10^{-1}	4.24×10^{-1}	7.8×10^{-1}	1.63×10^{-1}	3.23×10^{-2}	7.8×10^{-1}	5.64×10^{-1}	3.23×10^{-2}
HB	η_0 (Pa • s)	8.01×10^{-2}	1.42×10^{-1}	1.77×10^{-1}	5.73×10^{-2}	3.59×10^{-2}	8.43×10^{-1}	5.73×10^{-2}	1.54	1.28	1.71×10^{-4}	1.54	4.16	1.71×10^{-4}
	n	7.49×10^{-1}	6.12×10^{-1}	3.86×10^{-1}	1.06	1.12	9.22×10^{-1}	1.06	7.55×10^{-1}	1.48×10^{-1}	3.92	7.55×10^{-1}	5.1×10^{-1}	3.92
BH	η_0 (Pa•s)	2.29×10^{-2}	1.72×10^{-2}	8.60×10^{-3}	6.84×10^{-2}	5.25×10^{-2}	7.58×10^{-1}	6.84×10^{-2}	1.08	1.26×10^{-1}	1.71×10^{-2}	1.08	6.58×10^{-1}	1.71×10^{-2}
	σ_y (Pa)	1.03×10^{-1}	2.51×10^{-1}	6.19×10^{-1}	1.69×10^{-1}	$2. \times 10^{-1}$	1.78×10^{-2}	1.69×10^{-1}	5.61×10^{-2}	9.47×10^{-1}	7.48×10^{-1}	5.61×10^{-2}	1.08×10^{-1}	7.48×10^{-01}
CY	η_0 (Pa•s)	9.90	2.2×10^{1}	1.32×10^{3}	4.1×10^{2}	3.94×10^{2}	3.17×10^{3}	2.44×10^{-3}	4.94	7.25×10^{2}	1.58×10^{1}	2.03×10^{-1}	4.71×10^{1}	6.34×10^{-2}
	n	5.98×10^{-1}	4.23×10^{-1}	5.81×10^{-2}	-2.13×10^{-1}	-2.32×10^{-1}	2.27×10^{-1}	2.13×10^{-1}	8.42×10^{-1}	-2.19×10^{-1}	-1.41	-8.42×10^{-1}	3.01×10^{-1}	1.41
	η_∞(Pa•s)	-2.85×10^{-3}	-2.29×10^{-3}	5.01×10^{-3}	8.14×10^{-2}	6.45×10^{-2}	5.98×10^{-1}	8.14×10^{-2}	-2.78	4.48×10^{-2}	1.35×10^{-1}	-2.78	-1.61×10^{-2}	1.35×10^{-1}
	λ	1.97	1.84	3.44×10^{-4}	1.07×10^{-3}	1.48×10^{-3}	1.19	1.07×10^{-3}	3.64	1.82×10^{-2}	5.93×10^{-1}	3.64	1.67×10^{-1}	5.93×10^{-1}
	a	5.55×10^{-2}	7.26×10^{-2}	4.87×10^{-1}	3.10	3.54	5.57×10^{-2}	3.10	1.74×10^{-1}	2.45×10^{-1}	6.62×10^{-1}	1.74×10^{-1}	5.10×10^{-1}	6.62×10^{-1}
TC	η_0 (Pa•s)	1.73×10^{-2}	9.96×10^{-3}	-2.46×10^{-3}	6.81×10^{-2}	5.04×10^{-2}	7.32×10^{-1}	6.81×10^{-2}	9.78×10^{-1}	-1.15×10^{-1}	2.23×10^{-2}	9.78×10^{-1}	5.25×10^{-1}	2.23×10^{-2}
	$\dot{\gamma}_c$(s^{-1})	2.21×10^{-1}	2.13×10^{-1}	1.51×10^{1}	6.25×10^{3}	4.01×10^{4}	2.12×10^{-1}	1.60×10^{-4}	1.88×10^{-1}	8.89×10^{-1}	6.99×10^{4}	5.32	1.47×10^{-1}	1.43×10^{-5}
	σ_y (Pa)	2.69×10^{-2}	4.70×10^{-2}	5.00×10^{-1}	1.68×10^{-1}	2.16×10^{-1}	1.43×10^{-2}	1.68×10^{-1}	4.69×10^{-2}	6.17×10^{-1}	7.42×10^{-1}	4.69×10^{-2}	1.41×10^{-1}	7.42×10^{-1}
TCC	$\dot{\gamma}_c$	3.86×10^{-1}	7.49×10^{-1}	2.6×10^{1}	2.13×10^{-1}	9.68×10^{-1}	1.83×10^{-1}	4.69	1.73×10^{-1}	6.17	4.11×10^{-1}	5.78	1.52×10^{-1}	2.43
	$\dot{\gamma}_{cc}$(s^{-1})	1.35	8.68×10^{-1}	3.56×10^{-1}	3.21×10^{-1}	2.16×10^{-1}	7.07×10^{1}	3.12	1.48×10^{1}	3.93×10^{-1}	3.02	6.76×10^{-2}	7.75	3.31×10^{-1}
	σ_y (Pa)	1.9×10^{-1}	2.31×10^{-1}	4.80×10^{-1}	1.88×10^{-1}	3.16×10^{-1}	1.27×10^{-2}	1.88×10^{-1}	2.66×10^{-2}	6.32×10^{-1}	5.11×10^{-1}	2.66×10^{-2}	4.22×10^{-2}	5.11×10^{-1}
	k	-4.63×10^{-1}	-5.77×10^{-1}	2.57×10^{-1}	-1.29	-1.99	7.8×10^{-1}	-1.29	1.53	7.57×10^{-1}	-5.51×10^{-1}	1.53	6.92	-5.51×10^{-1}
Casson	η_0 (Pa•s)	1.83×10^{-2}	1.27×10^{-2}	2.53×10^{-3}	3.57×10^{-2}	2.46×10^{-2}	6.8×10^{-1}	3.57×10^{-2}	8.72×10^{-1}	2.32×10^{-2}	7.34×10^{-4}	8.72×10^{-1}	4.64×10^{-1}	7.34×10^{-4}
	σ_y (Pa)	3.33×10^{-2}	9.68×10^{-2}	5.44×10^{-1}	1.23×10^{-1}	1.54×10^{-1}	4.15×10^{-3}	1.23×10^{-1}	2.24×10^{-2}	8.07×10^{-1}	7.31×10^{-1}	2.24×10^{-2}	8.63×10^{-2}	7.31×10^{-1}
TEVP	η_s (Pa•s)	1.26×10^{-2}	9.57×10^{-3}	4.43×10^{-3}	7.71×10^{-2}	5.99×10^{-2}	5.55×10^{-1}	7.71×10^{-2}	-2.21×10^{-2}	2.38×10^{-3}	5.94×10^{-1}	-2.21×10^{-2}	3.37×10^{-2}	5.94×10^{-1}
	η_p (Pa•s)	3.12×10^{-2}	5.80×10^{-2}	2.58×10^{-1}	4.94	4.94	3.47×10^{-1}	4.94	1.8	1.3	-6.44×10^{-1}	1.8	6.96	-6.44×10^{-1}
	σ_y (Pa)	3.68×10^{-2}	3.64×10^{-2}	5.21×10^{-1}	3.90×10^{-1}	4.97×10^{-1}	1.35×10^{-2}	3.90×10^{-1}	2.86×10^{-2}	7.66×10^{-1}	8.00×10^{-1}	2.86×10^{-2}	3.80×10^{-2}	8.00×10^{-1}
	k^+	1.41	1.4	1.43	3.94×10^{-2}	4.75×10^{-2}	1.33	3.94×10^{-2}	1.32	1.06	1.62	1.32	1.58	1.62
	k^-	1.98×10^{-2}	4.97×10^{-2}	3.62×10^{-1}	1.82	1.81	1.48×10^{-1}	1.82	7.20×10^{-2}	8.3×10^{-1}	5.01×10^{-2}	7.20×10^{-2}	2.32×10^{-1}	5.01×10^{-2}
IKH	η_p (Pa•s)	2.08×10^{-2}	1.6×10^{-2}	4.43×10^{-3}	7.56×10^{-2}	5.86×10^{-2}	7.33×10^{-1}	7.56×10^{-2}	9.91×10^{-1}	1.46×10^{-2}	1.37×10^{-1}	9.91×10^{-1}	5.08×10^{-1}	1.37×10^{-1}
	c/q	1.95×10^{-1}	3.37×10^{-1}	1.02	1.18×10^{-1}	1.41×10^{-1}	3.06×10^{-2}	1.18×10^{-1}	1.12×10^{-1}	1.56	-5.97×10^{-1}	1.12×10^{-1}	4.58×10^{-2}	-5.97×10^{-1}
	k_1	-6.32×10^{-2}	-9.26×10^{-2}	1.26	1.55×10^{-3}	1.80×10^{-2}	-4.74×10^{-4}	1.55×10^{-3}	-7.51×10^{-4}	8.81×10^{-1}	1.37	-7.51×10^{-4}	1.68	1.37
	k_2	1.61	1.56	3.19×10^{-1}	1.70	1.69	1.79	1.70	1.72	8.50×10^{-1}	2.18×10^{-1}	1.72	-9.55×10^{-7}	2.18×10^{-1}
	k_3	4.16	4.66	-4.99×10^{-1}	3.5	5.72×10^{-1}	7.69×10^{-1}	3.5	2.14	-8.07×10^{-1}	1.4	2.14	8.06×10^{-2}	1.4

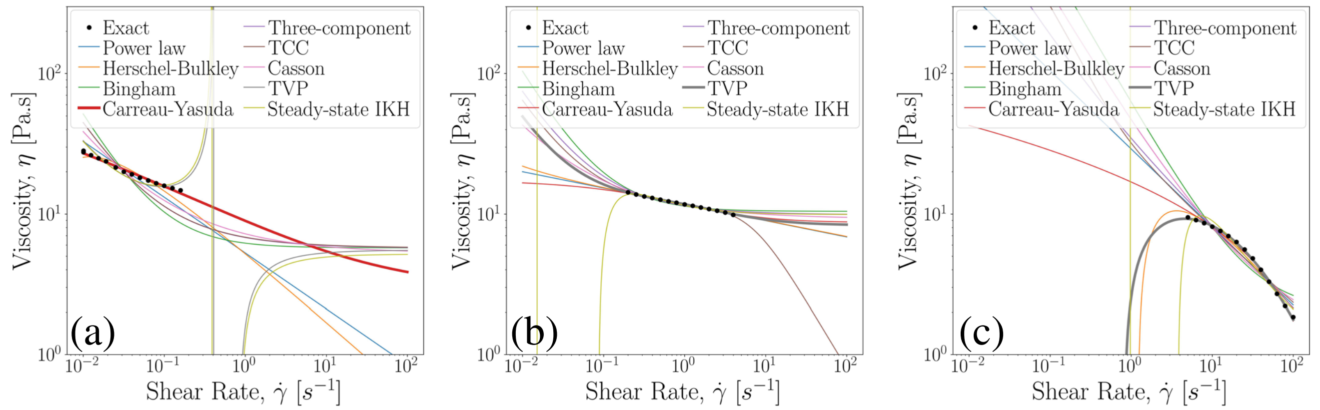

Fig. 6 Comparison of the model predictions for steady-state flow curve of the colloid/WLM mixture (sample no. 8), with different parameters recovered from different ranges of shear rates. Here, the initial data is equally divided into three regimes of **a** low, **b** intermediate, and **c** high shear rates and consequently fed into RhINN. For each curve, the thicker lines indicate the best model fit for that range

significant changes at different shear rate regimes, such as the Carbon black suspension (sample no. 3), model selection is relatively insensitive to the range of data provided.

Instead, one can choose to sample less number of imposed shear rates over the entire range of shear rates accessible with a rheometer/geometry. To test whether model predictions by the RhINN platform remain reasonable with loss of data, uniform sampling of initial data is provided for the model selection process, and results are presented in Fig. 7. For these test sets, input data are reduced to one-half, one-quarter, and one-eighth in size to find the extent to which RhINNs can select the appropriate model. For all three sets of data, not only is TVP the consistent recovered constitutive model, but also the model parameters remain reasonable and within a physical range for all models trained. These suggest that the most important factor in data acquisition is to cover an extended range of shear rates instead of favoring a specific region of shear rates.

Conclusion and future work

In this work, a rheology-informed neural network (RhINN) platform was developed and introduced to expeditiously select appropriate constitutive model(s) based on available experimental data. The proposed RhINN platform was found capable of identifying the best constitutive model for the experimental sample provided, spanning over six orders of magnitude in the shear rate and the shear stress. To demonstrate the capability of this platform, nine constitutive models representing a wide range of rheological phenomena were embedded in our RhINN platform, and a data set consisting of 13 steady-state flow curves from different complex fluids was provided. Our results indicate

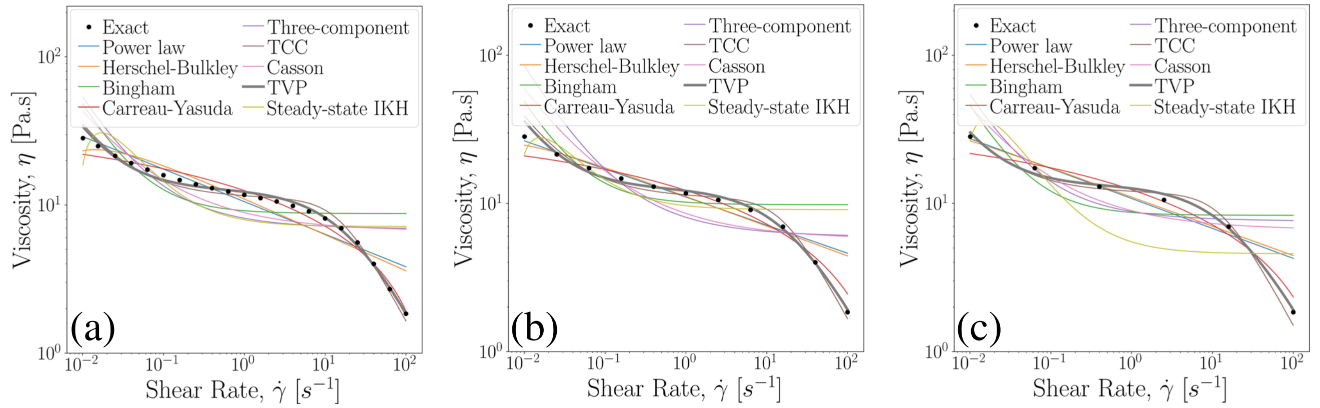

Fig. 7 Comparison of the model predictions for steady-state flow curve of the colloid/WLM mixture (sample no. 8), with different constitutive parameters recovered from different sizes of shear rate data. A reduced size of **a** one-half, **b** one-quarter, and **c** one-eighth of the entire initial data is used for the training process into the NN. For each curve, the thicker lines indicate the best model fit for that range

that the proposed RhINN architecture was capable of finding the appropriate model with the lowest number of fitting parameters for each data set with minimal user intervention. Thereupon, we showed that a uniform selection of a significantly reduced number of data over the entire accessible shear rates would not affect the RhINN's accuracy, as opposed to providing a specific range of data (and omitting the rest) which resulted in an erroneous model determination. Our findings suggest that data-driven model selection for rheological constitutive models can be reliably performed using RhINN platforms.

Nevertheless, there are other modern optimization methods based on adaptive parallel tempering (Armstrong et al. 2016) or Bayesian inference (Freund and Ewoldt 2015) that one may adapt to perform the model selection task. In all fairness, our developed platform, at its current state, does not claim to be the fastest (or the most accurate) option in one's arsenal for the model selection task, as explained in the "RhINN: convergence and benchmark" section. There are a few issues, e.g., unphysical parameter recovery and training runtime, that we need to address. However, we posit that RhINNs, with their powerful features such as automatic differentiation, straightforward scale-up, and robustness in unconstrained optimization, can complement other model selection techniques. Future work can include the implementation of fully resolved tensorial forms of non-linear viscoelastic models and various functional descriptions of deformation tensors within the neural network. By doing so, one can also recover the most appropriate non-linear rheological features of a system from a limited number of experimental data.

Acknowledgements The authors would like to thank Prof. George Em. Karniadakis for fruitful discussions about method development.

Funding The authors acknowledge the support of Northeastern University's Spark Fund program, as well as the National Science Foundation's DMREF #2118962 award.

Declarations

The authors declare no competing interests.

References

Armstrong MJ, Beris AN, Rogers SA, et al. (2016) Dynamic shear rheology of a thixotropic suspension: comparison of an improved structure-based model with large amplitude oscillatory shear experiments. J Rheol 60:433-450. https://doi.org/10.1122/1.4943986

Armstrong MJ, Beris AN, Wagner NJ (2017) An adaptive parallel tempering method for the dynamic data-driven parameter estimation of nonlinear models. AIChE J 63:1937-1958. https://doi.org/10.1002/aic.15577

Arora S, Laha A, Majumdar A, et al. (2017) Prediction of rheology of shear thickening fluids using phenomenological and artificial neural network models. Korea-Australia Rheology Journal 29:185-193. https://doi.org/10.1007/s13367-017-0019-x

Arulkumaran K, Deisenroth M P, Brundage M, et al. (2017) Deep reinforcement learning: a brief survey. IEEE Signal Proc Mag 34:26-38. https://doi.org/10.1109/MSP.2017.2743240

Barthés-Biesel D, Acrivos A (1973) The rheology of suspensions and its relation to phenomenological theories for non-Newtonian fluids. Int J Multiphase Flow 1:1-24. https://doi.org/10.1016/0301-9322(73)90002-5

Bingham EC (1916) An investigation of the laws of plastic flow. Bulletin of the Bureau of Standards 13:309-353

Bird RB (1965) Experimental tests of generalised Newtonian models containing a zero-shear viscosity and a characteristic time. The Canadian J Chemical Eng 43:161-168. https://doi.org/10.1002/cjce.5450430402

Bird RB, Armstrong RC, Hassager O (1987) Dynamics of polymeric liquids, vol 1. Fluid Mechanics. Wiley-Interscience

Blechschmidt J, Ernst OG (2021) Three ways to solve partial differential equations with neural networks – a review. GAMM-Mitteilungen:44. https://doi.org/10.1002/gamm.202100006

Brunton SL, Noack BR, Koumoutsakos P (2020) Machine learning for fluid mechanics. Ann Rev Fluid Mech 52:477-508. https://doi.org/10.1146/annurev-fluid-010719-060214

Caggioni M, Trappe V, Spicer PT (2020) Variations of the Herschel-Bulkley exponent reflecting contributions of the viscous continuous phase to the shear rate-dependent stress of soft glassy materials. J Rheol 64:413-422. https://doi.org/10.1122/1.5120633

Cai S, Mao Z, Wang Z et al (2022) Physics-informed neural networks (PINNs) for fluid mechanics: a review. Acta Mechanica Sinica. https://doi.org/10.1007/s10409-021-01148-1

Cardiel JJ, Dohnalkova AC, Dubash N, et al. (2013) Microstructure and rheology of a flow-induced structured phase in wormlike micellar solutions. Proc Natl Acad Sci U S A, vol 110. https://doi.org/10.1073/pnas.1215353110

Cuomo S, Di Cola VS, Giampaolo F et al (2022) Scientific machine learning through physics-informed neural networks: where we are and what's next. arXiv:220105624

Dagès N, Lidon P, Jung G, et al. (2021) Mechanics and structure of carbon black gels under high-power ultrasound. J Rheol 65:477-490. https://doi.org/10.1122/8.0000187

Dimitriou CJ, McKinley GH (2014) A comprehensive constitutive law for waxy crude oil: a thixotropic yield stress fluid. Soft Matter 10:6619-6644. https://doi.org/10.1039/c4sm00578c

Dunstan DE (2019) The viscosity-radius relationship for concentrated polymer solutions. Scientific Reports:9. https://doi.org/10.1038/s41598-018-36596-6

Freund JB, Ewoldt RH (2015) Quantitative rheological model selection: good fits versus credible models using Bayesian inference. J Rheol 59:667-701. https://doi.org/10.1122/1.4915299

Fuchs M, Ballauff M (2005) Nonlinear rheology of dense colloidal dispersions: a phenomenological model and its connection to mode coupling theory. Colloids Surf A Physicochem Eng Aspect 270-271:232-238. https://doi.org/10.1016/j.colsurfa.2005.06.017

Gao Y, Liu H, Wang X, et al. (2022) On an artificial neural network for inverse scattering problems. J Comput Phys 448:110-771. https://doi.org/10.1016/j.jcp.2021.110771

Herschel WH, Bulkley R (1926) Konsistenzmessungen von gummibenzollösungen. Kolloid-Zeitschrift 39:291-300. https://doi.org/10.1007/BF01432034

Huang B, Li X, Fu C, et al. (2019) Study rheological behavior of polymer solution in different-medium-injection-tools. Polymers, vol 11. https://doi.org/10.3390/polym11020319

Joshi YM (2022) Thixotropy, nonmonotonic stress relaxation, and the second law of thermodynamics. J Rheol 66:111-123. https://doi.org/10.1122/8.0000363

Karniadakis GE, Kevrekidis IG, Lu L, et al. (2021) Physics-informed machine learning. Nature Rev Phys 3:422-440. https://doi.org/10.1038/s42254-021-00314-5

Kohonen T (2001) Self-organizing maps, vol 30. Springer, Berlin. https://doi.org/10.1007/978-3-642-56927-2

Larson RG (2015) Constitutive equations for thixotropic fluids. J Rheol 59:595-611. https://doi.org/10.1122/1.4913584

Larson RG, Wei Y (2019) A review of thixotropy and its rheological modeling. J Rheol 63:477-501. https://doi.org/10.1122/1.5055031

Macosko CW (1994) Rheology: principles, measurements, and applications. VCH

Mahmoudabadbozchelou M, Jamali S (2021) Rheology-informed neural networks (RhINNs) for forward and inverse metamodelling of complex fluids. Scientific Reports, vol 11. https://doi.org/10.1038/s41598-021-91518-3

Mahmoudabadbozchelou M, Caggioni M, Shahsavari S, et al. (2021) Data-driven physics-informed constitutive metamodeling of complex fluids: a multifidelity neural network (MFNN) framework. J Rheol 65:179-198. https://doi.org/10.1122/8.0000138

Mahmoudabadbozchelou M, Karniadakis GE, Jamali S (2022) nn-PINNs: non-Newtonian physics-informed neural networks for complex fluid modeling. Soft Matter 18:172-185. https://doi.org/10.1039/D1SM01298C

Morrison FA (2001) Understanding rheology, Vol 1. Oxford University Press, New York

Nair SK, Basu S, Sen B, et al. (2019) Colloidal gels with tunable mechanomorphology regulate endothelial morphogenesis. Scientific Reports, vol 9. https://doi.org/10.1038/s41598-018-37788-w

Penwarden M, Zhe S, Narayan A, et al. (2022) Multifidelity modeling for physics-informed neural networks (PINNs). J Comput Phys 451:110-844. https://doi.org/10.1016/j.jcp.2021.110844

Raissi M, Perdikaris P, Karniadakis G (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686-707. https://doi.org/10.1016/j.jcp.2018.10.045

Raissi M, Yazdani A, Karniadakis GE (2020) Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. Science 367:1026-1030. https://doi.org/10.1126/science.aaw4741

Ritto T, Rochinha F (2021) Digital twin, physics-based model, and machine learning applied to damage detection in structures. Mech Syst Signal Process 155:107-614. https://doi.org/10.1016/j.ymssp.2021.107614

Soleymanzadeh A, Gahrooei HRE, Joekar-Niasar V (2018) A new empirical model for bulk foam rheology. J Energy Resources Technol, vol 140. https://doi.org/10.1115/1.4038385

De Souza Mendes PR (2011) Thixotropic elasto-viscoplastic model for structured fluids. Soft Matter 7:2471. https://doi.org/10.1039/c0sm01021a

Sun B, Barnard AS (2019) Visualising multi-dimensional structure/property relationships with machine learning. J Phys Materials 2:034,003. https://doi.org/10.1088/2515-7639/ab0faa

Sun T, Wang J, Li X, et al. (2013) Comparative evaluation of support vector machines for computer aided diagnosis of lung cancer in ct based on a multi-dimensional data set. Comput Methods Prog Biomed 111:519-524. https://doi.org/10.1016/j.cmpb.2013.04.016

Wang JX, Wu JL, Xiao H (2017) Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. Phys Rev Fluids 2:034,603. https://doi.org/10.1103/PhysRevFluids.2.034603

Zhu Q, Liu Z, Yan J (2021) Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. Comput Mech 67:619-635. https://doi.org/10.1007/s00466-020-01952-9