TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Chandra, A., Kapoor, T., Daniels, B., Curti, M., Tiels, K., Tartakovsky, D. M., & Lomonova, E. A. (2025). Generalizable models of magnetic hysteresis via physics-aware recurrent neural networks. *Computer Physics Communications*, *314*, Article 109650. https://doi.org/10.1016/j.cpc.2025.109650

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

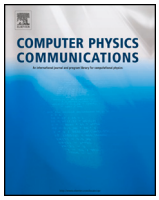
Computer Physics Communications 314 (2025) 109650



Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



Computational Physics



Generalizable models of magnetic hysteresis via physics-aware recurrent neural networks

Abhishek Chandra [a,*,1], Taniya Kapoor [b,1], Bram Daniels [a], Mitrofan Curti [a], Koen Tiels [c], Daniel M. Tartakovsky [d], Elena A. Lomonova [a]

[a] Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, 5600 MB, the Netherlands
[b] Department of Engineering Structures, Delft University of Technology, Delft, 2628CN, the Netherlands
[c] Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, 5600 MB, the Netherlands
[d] Department of Energy Science and Engineering, Stanford University, California, 94305, USA

ARTICLE INFO

The review of this paper was arranged by Prof. Andrew Hazel

Keywords:
Hysteresis
Magnetic material
Material modeling
Generalization
Physics-aware recurrent neural networks
Nonlinear dynamics

ABSTRACT

Hysteresis is a ubiquitous phenomenon in magnetic materials; its modeling and identification are crucial for understanding and optimizing the behavior of electrical machines. Such machines often operate under uncertain conditions, necessitating modeling methods that can generalize across unobserved scenarios. Traditional recurrent neural architectures struggle to generalize hysteresis patterns beyond their training domains. This paper mitigates the generalization challenge by introducing a physics-aware recurrent neural network approach to model and generalize the hysteresis manifesting in sequentiality and history-dependence. The proposed method leverages ordinary differential equations (ODEs) governing the phenomenological hysteresis models to update hidden recurrent states. The effectiveness of the proposed method is evaluated by predicting generalized scenarios, including first-order reversal curves and minor loops. The results demonstrate robust generalization to previously untrained regions, even with noisy data, an essential feature that hysteresis models must have. The results highlight the advantages of integrating physics-based ODEs into recurrent architectures, including superior performance over traditional methods in capturing the complex, nonlinear hysteresis behaviors in magnetic materials. The codes and data related to the paper are at github.com/chandratue/HystRNN.

1. Introduction

Magnetic hysteresis is a widely observed phenomenon in ferromagnetic and ferrimagnetic materials, where the change in magnetization response lags behind variations in the applied magnetic field. Specifically, hysteresis is characterized by a delay in the magnetic flux density (B) to changes in the applied magnetic field strength (H), exhibiting history dependence and nonlinearity [1,2]. The relationship between B and H fields is represented by a hysteresis curve C, also known as the B-H curve (Fig. 1), which assists in quantifying hysteresis and the governing magnetization process during alterations in H. The hysteresis curve, when closed, forms a loop known as the hysteresis loop, which offers several insights into material behavior [3]; for instance, the area of the hysteresis loop signifies the energy dissipated [4] as heat during each cycle of magnetization and demagnetization [5].

Accurate hysteresis modeling is pivotal in understanding and estimating the operational efficiency of magnetic material-based devices. For instance, the efficiency of electrical machines is intrinsically linked to the precise modeling of the hysteresis characteristics exhibited by the material employed [6,7]. A robust hysteresis model would avoid the costly manufacturing of multiple prototypes. Mathematically, the primary objective of hysteresis modeling is to predict the sequence of B values corresponding to a given sequence of H values. However, the B-H relationship defies the mathematical definition of a single-valued function. Consequently, conventional function approximation techniques are unsuitable for modeling hysteresis as a function with domain H and codomain B [8].

Traditionally, phenomenological models are employed for magnetic hysteresis modeling to establish relations between the H and B fields [9]. Notable phenomenological models include the Preisach [10], Jiles–Atherton [11], and Bouc–Wen model [12,13]. However, generalizing

* Corresponding author.
E-mail address: a.chandra@tue.nl (A. Chandra).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.cpc.2025.109650
Received 7 June 2024; Received in revised form 25 February 2025; Accepted 25 April 2025
Available online 5 May 2025
0010-4655/© 2025 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

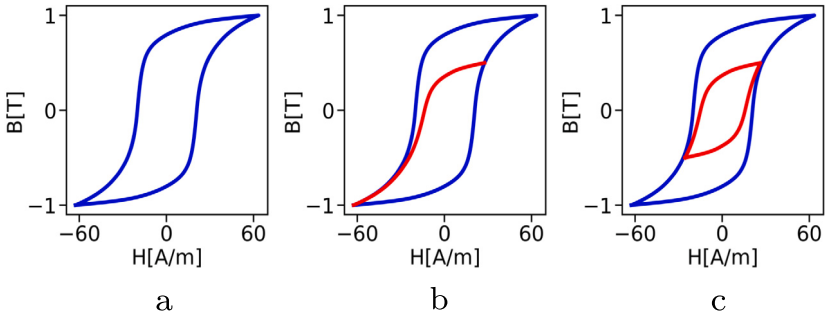


Fig. 1. B-H magnetic hysteresis curves: (a) Major loop (b) First-order reversal curve (Red) (c) Minor loop (Red). The Blue and Red curves represent the data on which the recurrent neural models are trained and tested respectively. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

these models across disciplines, fitting them to experimental data, and integrating them into other mathematical models pose challenges [14], such as sophisticated optimization techniques and increased computational burden [15].

To mitigate the limitations of phenomenological models, feed-forward neural networks (FFNNs) have been employed for modeling magnetic hysteresis [16,17,5,18]. However, owing to the absence of a functional relationship between B and H fields, the traditional FFNN approach with input H and output B is inadequate. Instead, studies [17,5] propose using several magnetization states as inputs to the FFNN to train it. However, this approach is suboptimal due to two notable limitations. First, it lacks the sequential information and fails to capture interdependencies among output values, agnostic to the underlying physics of the problem. Second, this strategy exhibits limitations in generalizing beyond training data [19], hindering their broader applicability in tasks that demand reliable performance in unseen scenarios.

Recurrent neural network (RNN) and its advanced variants pose a natural framework for modeling the sequential hysteretic nature. Consequently, several studies have utilized its sequential characteristics for magnetic hysteresis modeling, including but not limited to [8,20,21]. Still, the models employing traditional RNNs, gated recurrent unit (GRU) [22], and long-short-term memory (LSTM) [23] exhibit limitations in generalizing effectively to unseen H variations [24]. Even though these recurrent networks model the underlying relationship and predict hysteresis loops accurately in an interpolation task, achieving robust generalization remains an open problem [25]. This work posits that an optimal neural hysteresis model must demonstrate reasonable accuracy in generalization, effectively predicting B sequences for unseen H sequences.

A possible approach to achieve generalization is to enforce the recurrent architecture to incorporate the underlying dynamics. An efficient way to represent sequential dynamics involves representation through ordinary differential equations (ODEs) or dynamical systems, excelling in modeling diverse, intricate and nonlinear phenomena [26,27]. This work aims to employ a system of ODEs to update the hidden states of the recurrent architecture to encapsulate physical attributes of the underlying magnetic material. Recently, ODE based recurrent architectures have shown significant success in machine learning and artificial intelligence by handling the exploding and vanishing gradient problem effectively with high expressivity [28–30].

This work proposes a physics-aware recurrent neural network, hysteresis recurrent neural network (HystRNN). The proposed network embeds the hysteretic nature within the ODE formulation by leveraging phenomenological differential hysteresis models. Phenomenological models like Bouc–Wen [31] and Duhem [32] utilize the absolute value function to represent the underlying dynamics. By incorporating the absolute value function, the proposed method aims to effectively capture the shape of the hysteresis loop, and facilitate robust generalization, preserving symmetry and structure [33]. The main contributions of this paper are summarized as follows,

- To the best of the authors' knowledge, this is the first work to advocate neural differential equations for modeling and generalizing magnetic hysteresis.
- This work proposes a physics-aware recurrent neural network, HystRNN, merging the potentials of recurrent neural architectures with traditional differential hysteresis models.
- HystRNN outperforms the recurrent and gated neural architectures in generalizing the hysteresis model to unseen magnetic field variations.
- The proposed method is empirically shown to model hysteresis and generalize to unseen tasks while handling noise showcasing robustness.

The rest of the paper is structured as follows. Section 2 discusses related works to this paper in the literature. Section 3 presents the challenge mitigated in this paper and discusses generalization in the context of magnetic hysteresis modeling. Section 4 presents the proposed method and the corresponding ODEs. Section 5 validates the proposed method through a series of numerical experiments. Finally, Section 6 collates this study's key findings and implications.

2. Related works

This section outlines the relevant studies concerning recurrent neural architectures for modeling magnetic hysteresis, ODE-based recurrent architectures, and generalization studies within neural differential equations.

Several studies have utilized the sequential characteristics of recurrent neural architectures for modeling magnetic hysteresis. In particular, [34–36], among others, used recurrent neural networks to model the nonlinearity. Additionally, works have employed advanced gated variants of RNNs to model the nonlinearity, such as LSTM-based networks [37–39] and GRU-based networks [40–42]. This work also employs a recurrent-based architecture to model magnetic hysteresis. However, the aforementioned studies are confined to prediction in the training domain, whereas this work focuses on generalization by predicting in out-of-domain scenarios.

RNN architectures incorporating ODEs and dynamical systems for hidden state updates have been utilized for diverse engineering applications [43–45]. These physics-aware networks have further been proved to mitigate the exploding and vanishing gradient problem [29] and be universal function approximators [46].

ODE-based recurrent networks have exhibited the potential to generalize the learned model to out-of-distribution prediction scenarios. [47,48] employed a second-order ODE-based recurrent network to extend the applicability of physics-informed machine learning by extrapolating the solutions of partial differential equations. However, this work aims to predict trajectories of the hysteresis dynamics without relying on retraining or transfer learning methodologies [49,50] to predict the quantity of interest in a generalized domain.

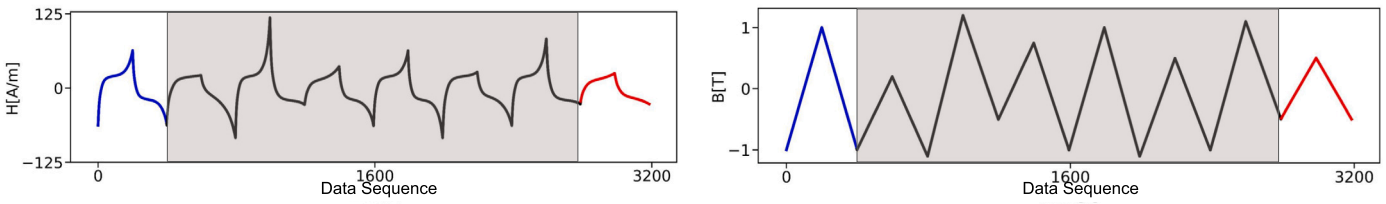


Fig. 2. Sequence of magnetization and demagnetization for a magnetic material: (a) H sequence (b) B sequence. The Blue and Red curves represent the data on which the recurrent methods are trained and tested respectively. The Black curve behind the shaded region represents the dynamics unseen to the recurrent methods.

The next section presents the notion of generalization in modeling magnetic hysteresis, introducing major, minor and first-order reversal curves.

3. Generalization of hysteresis models

Traditional supervised machine learning methodologies employed to model magnetic hysteresis train the model for input-output data pairs $(H_i, B_i) \in C_1$, where $1 \leq i \leq N$, $i \in \mathbb{Z}$ and N is the number of training samples. The trained model is then tested on $(H_k, B_k) \in C_2$, where $1 \leq k \leq M$, $k \in \mathbb{Z}$ and M is the number of testing samples. Traditionally, $C_2 \subset C_1$, with $H_i \neq H_k$. However, this prediction reduces to an interpolation task [41]. In contrast, this work focuses on training the model for $(H_i, B_i) \in C_1$ and predicting a hysteresis trajectory for $(H_k, B_k) \in C_2$, where $C_2 \nsubseteq C_1$, and $C_2 \cap C_1 = \phi$. Here, ϕ denotes the null set. Precisely, the models are trained on the major loop (C_{major}) as shown in Fig. 1a. Then, the trained model is tested for two different scenarios. First the FORCs (C_{FORC}) shown by the Red curve in Fig. 1b and second the minor loop (C_{minor}) presented by the Red curve in Fig. 1c.

Modeling FORCs and minor hysteresis loops play a significant role in analyzing magnetic materials. FORC modeling reveals intricate interactions, enabling the differentiation between magnetization components that can be reversed and those that cannot. This knowledge is pivotal in optimizing magnetic devices like memory and sensor technologies [51–54]. Minor hysteresis loop modeling complements this by providing insights into localized variations in magnetic behavior.

The notion that simulating C_{FORC} and C_{minor} entails to a generalization task is further described using Fig. 2. In Fig. 2a and 2b, the sequence of H and B fields are shown, respectively. The Blue curve represents the training data (H vs B is C_{major}), and the Red curve represents the region in which the prediction is sought (C_{minor} in this case). The Black curve behind the shaded region signifies the history that the material has gone through, which is the series of magnetization and demagnetization that is unknown while testing. Hence, this task amounts to extrapolation or predicting in a generalized scenario.

The next section presents the proposed method and the ODEs employed to predict the generalized scenario.

4. Method

The proposed method, HystRNN, utilizes a recurrent structure akin to RNNs, with the difference being in the hidden state update. HystRNN draws similarities to deep learning architectures that employ the concept of neural differential equations [26] to model the dynamics of hidden states. Neural differential equations, by definition, model the evolution of the hidden states through differential equations. Contrary to traditional deep learning architectures that model the hidden state update discretely, neural differential equations employ a differential equation-based strategy to evolve the hidden states continuously, preserving long-term dependencies and effectively mitigating the vanishing and exploding gradient problem typical with recurrent neural network-based methods [28]. The computed hidden states enhance the training stability by remaining within bounds and maintaining a robust representation of sequential data.

In HystRNN, the characteristics of neural differential equations are incorporated through motivations from the differential models of hysteresis. Examples of such phenomenological models include but are not limited to the Bouc–Wen [12,13] and Duhem models [32]. Instead of using recurrent and gated neural architectures to model hysteresis, HystRNN employs neural differential equations to update the hidden states. In particular, an important characteristic of these models is the incorporation of an absolute value function in the hidden state update to model the hysteresis nonlinearity. These absolute valued components play a crucial role in capturing hysteretic characteristics by allowing the model to account for different responses during magnetization and demagnetization and accounting for the effects of history on the system's behavior. Including the phenomenological model-inspired ODE containing the absolute valued terms enhances the ability of the model to capture the intricate dynamics of hysteresis and provides a more realistic representation of the observed phenomena.

The method is graphically illustrated in Fig. 3. The approach involves two inputs, H and B_{-1}, mapped to B, where B_{-1} denotes the previous B state. The modeling process begins by collecting N_e number of measured data points for the sequence $(H_i, B_i) \in C_{\text{major}}$ in the data space, where $1 \leq i \leq N_e$, and $i \in \mathbb{Z}$. These data are collected by first uniformly sampling the desired H field, which assists in obtaining the information between H_{-1} and H. The uniformly sampled H field is applied to the material (or its physical model) to generate the B field. In this paper, the B field does not follow uniform stepping owing to the nonlinear hysteretic property of magnetic materials. Subsequently, $\mathbf{u} := (H_j, B_k)$ and B_j are taken as the input and output of HystRNN, respectively, where $2 \leq j \leq N_e$, $1 \leq k \leq N_e - 1$, $k = j - 1$, and $j, k \in \mathbb{Z}$. The number of training points is denoted by $N = N_e - 1$. While sharing certain similarities with some FFNN architectures employed for modeling hysteresis, this training approach diverges by incorporating a recurrent relationship that captures longer-time dynamics and output dependencies, which are absent in FFNNs. Next, the hidden states of HystRNN are updated in the latent space through the following second-order ODE,

$$\begin{aligned}\mathbf{y}'' = \sigma_1\left(\mathbf{W}_1\mathbf{y} + \boldsymbol{\mathcal{W}}_1\mathbf{y}' + \mathbf{V}_1 \circ \mathbf{u} + \mathbf{b}_1\right) + \sigma_2\big(\mathbf{W}_2|\mathbf{y}|^2 + \boldsymbol{\mathcal{W}}_2|\mathbf{y}'|^2 \\ + \mathbf{V}_2 \circ |\mathbf{u}|^2 + \mathbf{b}_2\big).\end{aligned} \tag{1}$$

Here, the hidden state of the HystRNN is denoted by $\mathbf{y} = \mathbf{y}(t) \in \mathbb{R}^m$. $\mathbf{y}'$ indicates a derivative of $\mathbf{y}$ with respect to t, while $\mathbf{y}''$ indicates a second-order derivative of $\mathbf{y}$ with respect to t. $\mathbf{W}_1, \mathbf{W}_2, \boldsymbol{\mathcal{W}}_1, \boldsymbol{\mathcal{W}}_2 \in \mathbb{R}^{m\times m}$, and $\mathbf{V}_1, \mathbf{V}_2 \in \mathbb{R}^{m\times N}$ are the weight matrices, and t corresponds to the sequence index at which the training data has been collected. $\mathbf{u} = \mathbf{u}(t) \in \mathbb{R}^{N\times 2}$ is the input to HystRNN. The dimension of $\mathbf{u}$ is N × 2, with N rows and 2 columns. The N rows represent the number of data samples, and 2 columns consist of H_j data in the first column and B_k data in the second column. The notation $\circ$ denotes that the product of the corresponding weight matrix and $\mathbf{u}$ is performed column-wise, and the columns of the resulting matrices are summed to obtain the product. The bias vectors are denoted by $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^m$. The activation functions $\sigma_1 : \mathbb{R} \mapsto \mathbb{R}$ and $\sigma_2 : \mathbb{R} \mapsto \mathbb{R}$ are taken to be $\sigma_1(u) = \sigma_2(u) = \tanh(u)$.

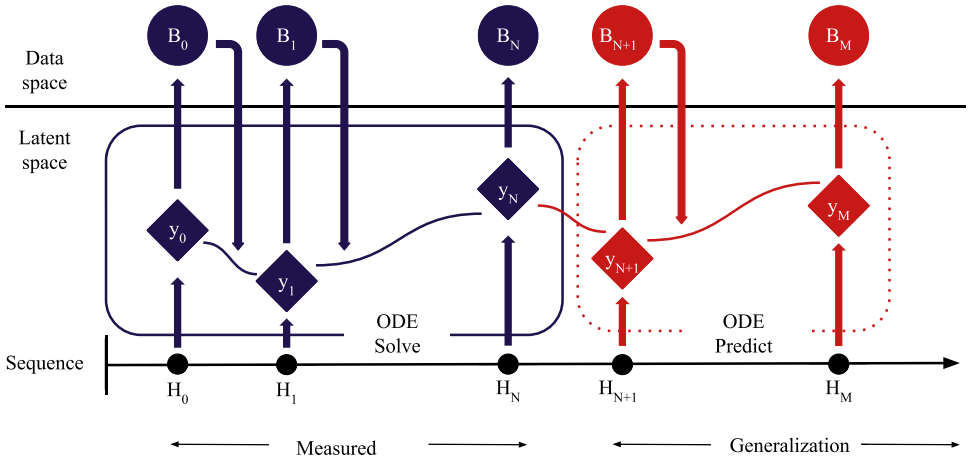


Fig. 3. Proposed recurrent neural network, HystRNN, for modeling magnetic hysteresis.

Other activation functions constraining the output to specific ranges as the tanh activation function can also be considered [55]. Furthermore, the absolute value function is mathematically defined as

$$|u| = \begin{cases} u & \text{if } u \geq 0, \\ -u & \text{if } u < 0. \end{cases}$$

The definition of the absolute value function remains consistent throughout the paper.

The proposed hidden state update (1) leverages a differential model. One of the phenomenological differential models characterizing hysteresis is the Duhem model [32]. The Duhem model is used to describe hysteresis by establishing a rate-dependent differential relationship between input (H) and output (B) as follows,

$$\mathrm{B}' = \alpha|\mathrm{H}'|\mathrm{H} - \beta|\mathrm{H}'|\mathrm{B} + \gamma\mathrm{H}', \tag{2}$$

where α, β, and γ represent parameters of the model. The model captures hysteresis characteristics, such as memory effects and nonlinearity making it a foundational model in various physical systems. Inspired by this, the proposed formulation (1) extends the Duhem model by introducing a second-order differential equation incorporating linear and nonlinear terms. Specifically, it includes weighted contributions of the state $\mathbf{y}$, its derivative $\mathbf{y}'$, and the input $\mathbf{u}$, along with nonlinear terms $(|\mathbf{y}|^2, |\mathbf{y}'|^2, |\mathbf{u}|^2)$ to capture complex hysteretic behaviors. This formulation aims to enhance the model's ability to generalize hysteresis patterns, making it suitable for diverse applications in engineering and physics.

To solve the second order ODE (1), the substitution $\mathbf{z} = \mathbf{y}'(t) \in \mathbb{R}^m$ is performed, reducing (1) to a system of first-order ODEs

$$\begin{aligned} \mathbf{y}' &= \mathbf{z}, \\ \mathbf{z}' &= \sigma_1\left(\mathbf{W}_1\mathbf{y} + \boldsymbol{\mathcal{W}}_1\mathbf{z} + \mathbf{V}_1 \circ \mathbf{u} + \mathbf{b}_1\right) + \sigma_2\big(\mathbf{W}_2|\mathbf{y}|^2 + \boldsymbol{\mathcal{W}}_2|\mathbf{z}|^2 \\ &\quad + \mathbf{V}_2 \circ |\mathbf{u}|^2 + \mathbf{b}_2\big). \end{aligned} \tag{3}$$

Discretizing the system of ODEs (3) using an explicit scheme for $0 < \Delta t < 1$ leads to

$$\begin{aligned} \mathbf{y}_n &= \mathbf{y}_{n-1} + \Delta t\mathbf{z}_n, \\ \mathbf{z}_n &= \mathbf{z}_{n-1} + \Delta t\sigma_1\left(\mathbf{W}_1\mathbf{y}_{n-1} + \boldsymbol{\mathcal{W}}_1\mathbf{z}_{n-1} + \mathbf{V}_1 \circ \mathbf{u}_n + \mathbf{b}_1\right) \\ &\quad + \Delta t\sigma_2\left(\mathbf{W}_2|\mathbf{y}_{n-1}|^2 + \boldsymbol{\mathcal{W}}_2|\mathbf{z}_{n-1}|^2 + \mathbf{V}_2 \circ |\mathbf{u}_n|^2 + \mathbf{b}_2\right). \end{aligned} \tag{4}$$

Finally, to compute the output prediction $\hat{\mathrm{B}} \in \mathbb{R}^n$, a linear transformation $\hat{\mathrm{B}} = \mathcal{Q}\mathbf{y}_n$ and $\mathcal{Q} \in \mathbb{R}^{n\times m}$ is carried out following the solution of the system of two equations in (4). Once HystRNN is trained, it can be used to make predictions for unseen H sequences while testing the model, for instance, the sequence $\{\mathrm{H}_{N+1}, \ldots, \mathrm{H}_M\}$, as shown in Fig. 3.

The next section presents the numerical experiments to validate the proposed method.

5. Numerical experiments

A series of numerical experiments are presented, encompassing different generalization scenarios. The magnetic material, nonoriented electrical steel (NO27) is modeled for each experiment. The data for the modeled hysteresis loops are acquired using the Preisach model for an Epstein frame [56,57]. The Preisach model is adjusted to adhere to the IEC standard [58], and the core is assembled using 16 strips of NO27-1450H material. Further details about the Preisach model are provided in Appendix A. Additionally, it is imperative to normalize the data before feeding it into the deep learning models [59]. Measured data are normalized using the min-max scaling technique, as elucidated in Appendix B.

HystRNN is compared with traditional recurrent networks, RNN, LSTM, and GRU for generalization experiments owing to their intrinsic sequentiality and memory dependence characteristics. The methods are compared using four metrics. The first is the L2-norm, measuring the Euclidean distance between predicted and actual values. The second metric is the explained variance score, which indicates prediction accuracy, capturing variance proportion. The third metric is the maximum error, which detects significant prediction discrepancies as potential outliers. Finally, mean absolute error is chosen as the fourth metric assessing the average differences between predictions and actual values for overall precision. A lower L2-norm, maximum error, and mean absolute error coupled with higher explained variance signify a better performance. Metric expressions are detailed in Appendix C.

The common hyperparameters for all the experiments consist of an input size of 2, a single hidden layer with a dimension of 32, and an output size of 1. The optimization process involves the utilization of the Adam optimizer, with a learning rate of 0.01. Training is conducted for 10000 epochs, with a batch size of 1. The hyperparameter Δt is chosen to be 0.05. The hyperparameter Δt used in HystRNN refers specifically to the time-stepping in the numerical solution of the ODE and does not correspond to any physical time scale in the problem, as magnetic hysteresis is inherently a function of the input excitation history rather than explicit time evolution. Uniformity in hyperparameter settings is maintained across all experiments. Furthermore, the hyperparameters are the same for RNN, LSTM, and GRU to ensure fair comparisons.

For all the numerical experiments, the software and hardware environments used for performing the experiments are as follows: Ubuntu 20.04.6 LTS, Python 3.9.7, Numpy 1.20.3, Scipy 1.7.1, Matplotlib 3.4.3, PyTorch 1.12.1, CUDA 11.7, and NVIDIA Driver 515.105.01, i7 CPU, and NVIDIA GeForce RTX 3080. The codes and data related to the paper are at github.com/chandratue/HystRNN.

Table 1
The generalization performance assessed using the metrics: L2-norm relative error, explained variance error, maximum error, and mean absolute error for experiment 1, where $B_{max} = 1.7$ T. For these metrics, higher (respectively, lower) values are favored for (↑) (respectively, (↓)). The implication of arrows remains consistent for all the following Tables.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN
C_{FORC_1}	5.0204	0.8525	0.7764	**0.2198**	-0.0721	0.1081	0.2007	**0.8252**	5.3597	2.4089	2.0075	**1.2030**	2.9888	1.1967	1.5550	**0.6149**
C_{FORC_2}	6.4877	0.5255	0.4701	**0.3085**	-0.2545	0.1875	0.2395	**0.8844**	5.3484	1.8428	1.8177	**1.2371**	3.6038	0.9327	0.8723	**0.7613**
C_{minor_1}	5.3506	1.4382	1.8028	**0.0438**	-0.1013	0.0298	0.0776	**0.9839**	2.7641	1.7098	1.8925	**0.3108**	1.4877	0.7142	0.7797	**0.1258**
C_{minor_2}	12.3671	1.5785	2.0563	**0.0786**	-2.7046	0.0248	0.0673	**0.9661**	3.7491	1.5726	1.7486	**0.3630**	1.9703	0.6544	0.7341	**0.1450**

5.1. Train and test criteria

The models are trained only for the data on the major loop (represented by a Blue loop in Fig. 1a). They are tested for two different generalization tasks: predicting first-order reversal curves (FORCs, represented by the Red curve in Fig. 1b) and minor loops (represented by a Red loop in Fig. 1c). The trained models are tested in two distinct scenarios involving the prediction of two FORCs and two minor loops. A sequence length of 595 is chosen to train C_{major}. For FORC prediction, testing sequence of length 199 is utilized. The prediction of minor loops involves a testing sequence with a length of 399 each. The choice of sequence lengths depends on the data generated from the Preisach model.

During training both the data for H_j and B_j are available at sequence step j and are utilized to train the networks. Specifically, the network must map the data for H_j and B_{j-1} to B_j in a supervised manner. However, while testing, the objective is to infer the material's response, and hence, no prior knowledge of B_j is available for a given H_j. Hence, while testing, the learned representation is utilized to map H_j and B_{j-1} to B_j. This B_j is then used with H_{j+1} (which is available being the input field) to predict B_{j+1}. This process is repeated auto-regressively to predict the minor loops and FORCs. Concretely, the HystRNN model trained on C_{major} is evaluated on C_{FORC} and C_{minor}. This testing sequence is initiated with an input $(H_j, B_j) \in C_{\text{FORC/minor}}$, where both H_j, and B_j are provided and B_{j+1} is predicted. The output generated from this step, B_{j+1}, becomes the subsequent input along with H_{j+1}, the known magnetization for the following sequence. Such testing strategy is essential as practical scenarios lack prior knowledge about the B values on C_{FORC} or C_{minor}. Thus, the sole available information for generalization stems from the predicted solution in C_{FORC} or C_{minor}. All reported results are obtained for randomly selected excitation signals from the shaded region of Fig. 2, demonstrating the model's effectiveness in predicting C_{minor} and C_{FORC} over random input fields.

5.2. Generalization

Four numerical experiments, differing by the maximum permitted magnetic flux density of the electrical machine (B_{max}), are presented in this section to evaluate the generalization potential of HystRNN. The selection of different B_{max} values correspond to the material's specific usage context, demonstrating the proposed methodology's viability across a spectrum of electrical machines. Precisely, B_{max} values of 1.7 T, 1.5 T, 1.3 T, and 1.25 T are chosen.

Performing experiments and exploring the generalization capabilities of the model for varying B_{max} values is crucial for understanding and optimizing the performance and efficiency of different and diverse machines. For instance, machines requiring lower magnetic flux densities of $B_{max} = 1.25$ T are typically used as high-efficiency induction motors [60] in industrial settings for tasks like driving conveyor belts, pumps, and compressors. Meanwhile, high-performance applications, for instance, particle accelerators [61] and nuclear magnetic resonance [62], demand higher magnetic flux densities with $B_{max} = 1.7$ T for their operation. The experiments are presented for B_{max} in this spectrum, and the corresponding performance for generalized scenarios is examined. Furthermore, as B_{max} increases, the slope of the hysteresis curve becomes more steep, which increases the complexity for deep learning models [63,64].

For each of these experiments, a total of four cases are executed. The first and second test cases correspond to estimating FORC, denoted by C_{FORC_1} and C_{FORC_2} respectively. Two distinct FORCs are chosen to study the effect of the distance between the origin of the FORC and B_{max}. The third and fourth test cases are performed for predicting minor loops, denoted by C_{minor_1} and C_{minor_2} respectively. For testing the trained model on a random excitation corresponding to FORC or a minor loop, the excitation is re-applied to trace the major loop until the maximum B value from which the FORC or minor loop is to be predicted. The retracing of the major loop need not be trained or tested as the model used that data to train the model earlier, and the predictions for B fields are already available for the major loop.

Detailed performance metrics for HystRNN are outlined in Tables 1, 2, D.1, and D.2 corresponding to experiments 1, 2, 3, and 4, respectively. The tables also facilitate a comprehensive comparative analysis of RNN, LSTM, and GRU. The metrics notably emphasize the superior performance of HystRNN across all numerical experimentation scenarios.

5.3. Experiment 1

For the first experiment, $B_{max} = 1.7$ T. The origin of $C_{\text{FORC}_{1,2}}$ is taken to be 1.25 T and 0.5 T respectively. The chosen maximum B value of $C_{\text{minor}_{1,2}}$ is 1.25 T and 1.1 T. The predictions for the model, the ground truth, and the training data are presented in Fig. 4.

In Fig. 4, the top two rows display the predictions of $C_{\text{FORC}_{1,2}}$ respectively, wherein training exclusively occurs on C_{major}, indicated by the Blue color. Predictions are represented by the Black color, and ground truth is represented by the Red color. The colors are kept consistent for all the following experiments. The top two rows show that RNN (Fig. 4a, 4e), LSTM (Fig. 4b, 4f), and GRU (Fig. 4c, 4g) fail drastically to capture the shape of the FORC accurately. In contrast, HystRNN effectively captures the structure and symmetry of reversal curves as shown in Fig. 4d and 4h.

The last two rows show the prediction for minor loop $C_{\text{minor}_{1,2}}$. For this case also predictions from RNN (Fig. 4i, 4m), LSTM (Fig. 4j, 4n) and GRU (Fig. 4k, 4o) are inaccurate. Neither LSTM nor GRU could form a closed loop for the predicted trajectory, posing a major challenge to compute the energy loss without a closed region, as energy loss depends on the surface area of the hysteresis loop. In contrast, the proposed method HystRNN predicts the structure of the loop and efficiently models the minor loop (Fig. 4l, 4p). The accuracy of predicting C_{minor} is influenced by its similarity to the major loop. While model performance naturally decreases for more distinct patterns, HystRNN outperforms standard recurrent architectures in generalizing hysteresis behavior, even in low-data regimes.

5.4. Experiment 2

For the second experiment, $B_{max} = 1.25$ T. The origin of $C_{\text{FORC}_{1,2}}$ is taken to be 1.2 T and 1.0 T respectively. The chosen maximum B value of $C_{\text{minor}_{1,2}}$ is 1.0 T and 0.8 T. The predictions for the model, the ground truth, and the training data are presented in Fig. 5.

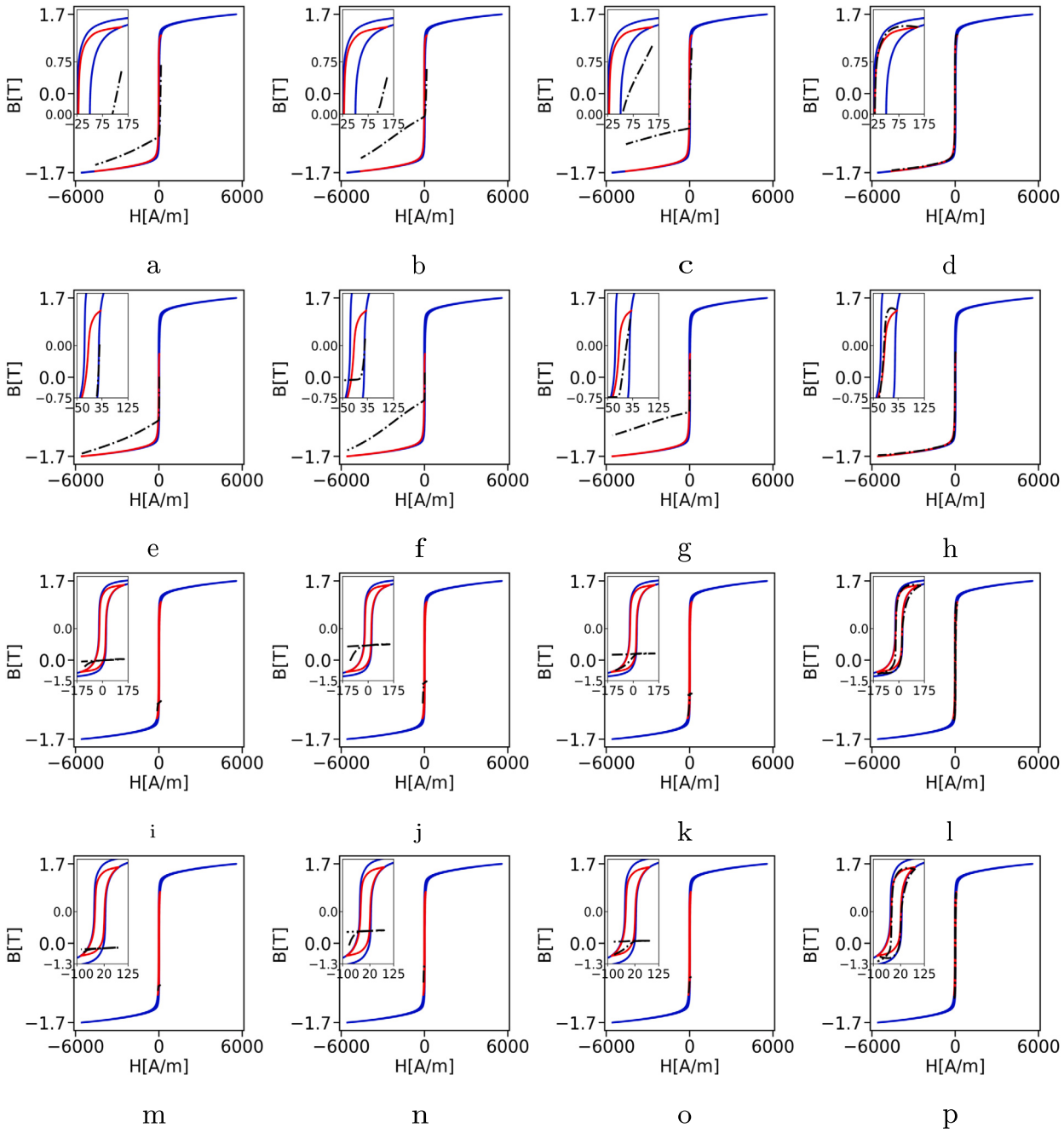


Fig. 4. Experimental vs predicted hysteresis trajectories for experiment 1, where $B_{max} = 1.7$ T. Columns represent solutions for different methods, RNN(4a, 4e, 4i, 4m), LSTM (4b, 4f, 4j, 4n), GRU (4c, 4g, 4k, 4o), HystRNN (4d, 4h, 4l, 4p). The Blue curve represents the training loop $\mathcal{C}_{\text{major}}$. The Red curve represents the ground truth for $\mathcal{C}_{\text{FORC/minor}}$, and the Black curve represents the model's prediction. Top two rows: predictions for $\mathcal{C}_{\text{FORC}_1}$ and $\mathcal{C}_{\text{FORC}_2}$ respectively. Bottom two rows: predictions for $\mathcal{C}_{\text{minor}_1}$ and $\mathcal{C}_{\text{minor}_2}$ respectively. The colors are used consistently for the following figures.

The top two rows of Fig. 5 present that RNN (Fig. 5a, 5e), LSTM (Fig. 5b, 5f) and GRU (Fig. 5c, 5g) fail to predict the trajectory of FORC by a huge margin. On the other hand, HysRNN shows close agreement with the ground truth for predicting $\mathcal{C}_{\text{FORC}_{1,2}}$ (Fig. 5d, 5h). Also, the prediction of HystRNN for $\mathcal{C}_{\text{FORC}_1}$ is slightly better than for $\mathcal{C}_{\text{FORC}_2}$, exemplifies that the model performs better when the origin of FORC is closer to B_{max}. A possible reason for this behavior could be the resemblance in the trajectories of $\mathcal{C}_{\text{major}}$ and a FORC from a higher origin value.

The last two rows of Fig. 5 present the predictions of $\mathcal{C}_{\text{minor}_{1,2}}$ respectively. For this case, RNN (Fig. 5i, 5m), LSTM (Fig. 5j, 5n) and GRU (Fig. 5k, 5o) almost form a loop-like shape; however, they are very off from compared to the ground truth. HystRNN, on the other hand, captures the loop shape efficiently, as presented in Fig. 5l and 5p.

In Fig. 5l and 5p, for the testing of minor loops, the excitation signals initiate from the lower left and increase toward the upper right. Once the upper right is reached, the excitation is applied continuously in the negative direction, decreasing back to the lower left. Due to the inherent approximation in the model and the challenges of generalization, slight discrepancies arise in the predicted trajectories, particularly in the lower left region, resulting in the observed cross-curve effect. This phenomenon results from minor deviations in the learned hysteresis behavior when extrapolating beyond the training domain.

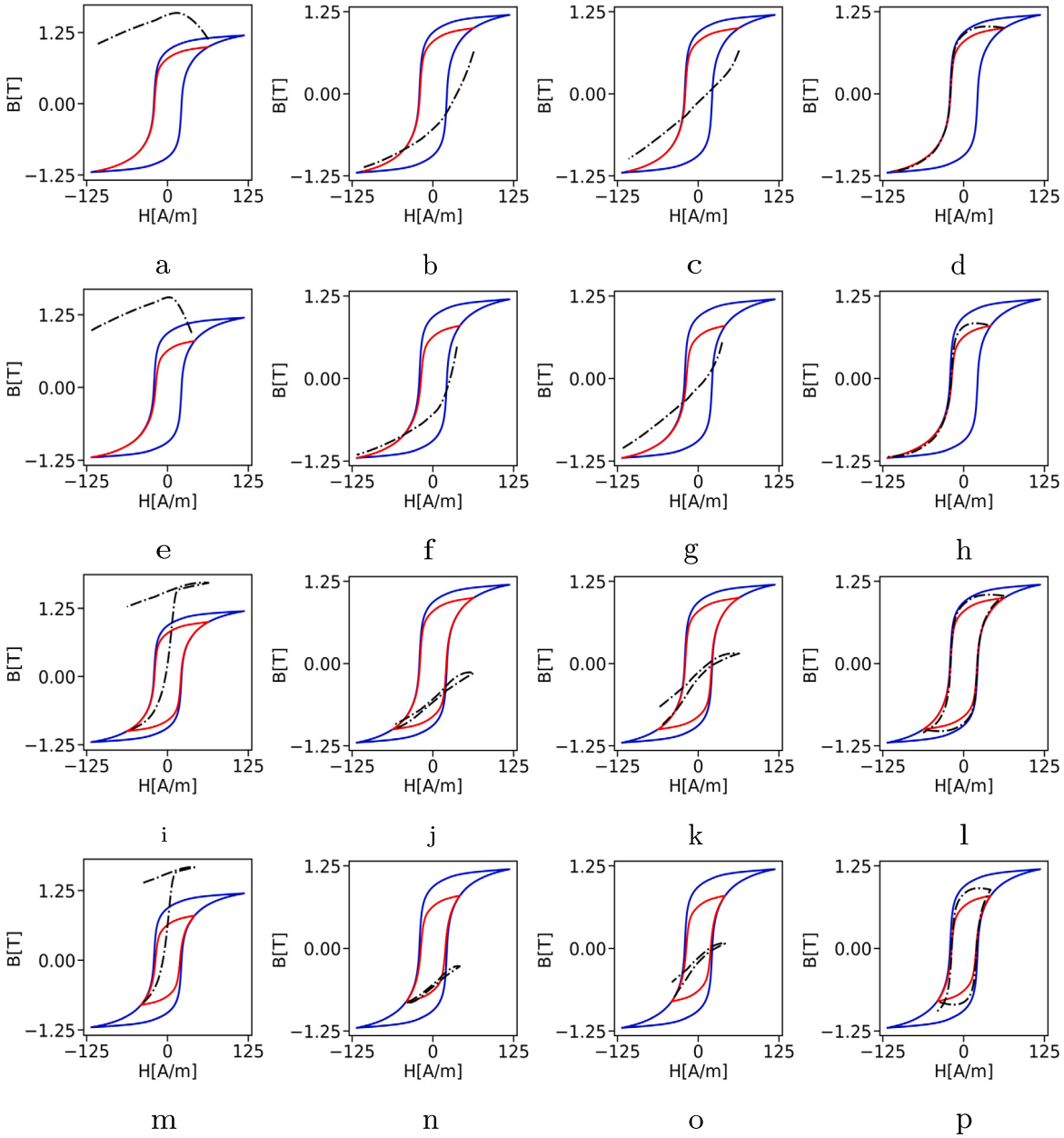


Fig. 5. Experimental vs predicted hysteresis trajectories for experiment 2, where $B_{max} = 1.25$ T. Columns represent solutions for different methods, RNN (5a, 5e, 5i, 5m), LSTM (5b, 5f, 5j, 5n), GRU (5c, 5g, 5k, 5o), HystRNN (5d, 5h, 5l, 5p). Top two rows: predictions for C_{FORC_1} and C_{FORC_2} respectively. Bottom two rows: predictions for C_{minor_1} and C_{minor_2} respectively.

Table 2
The generalization performance assessed using the metrics: L2-norm relative error, explained variance error, maximum error, and mean absolute error for experiment 2, where $B_{max} = 1.25$ T.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN
C_{FORC_1}	5.3234	1.2308	0.6863	**0.0109**	0.1625	0.3566	0.4230	**0.9891**	2.6134	1.4922	1.0994	**0.2370**	1.6253	0.6871	0.5433	**0.0563**
C_{FORC_2}	6.4035	0.9569	0.5098	**0.0115**	0.1576	0.3834	0.4693	**0.9924**	2.6084	1.2985	0.9055	**0.2520**	1.7484	0.5796	0.4587	**0.0583**
C_{minor_1}	7.5971	1.6295	0.8069	**0.0320**	0.1497	0.2977	0.3268	**0.9844**	2.3788	1.3344	0.9456	**0.1882**	1.4996	0.5974	0.4396	**0.0916**
C_{minor_2}	11.8293	2.1787	0.8859	**0.1283**	0.0616	0.2769	0.3091	**0.9267**	2.2669	1.1784	0.7925	**0.2923**	1.5243	0.5639	0.3653	**0.1443**

5.4.1. Ablation on train-test size

This section presents an ablation study for HystRNN varying the train and test size for the experiment $B_{max} = 1.25$ T. The experiment evaluates the accuracy of HystRNN as a function of N and M (the number of training and testing data points, respectively). Specifically, the hyperparameters N and M are varied systematically, and the corresponding errors are observed. The training data size is varied discretely across $\{300, 600, 900, 1200\}$, while the testing data size for FORCs is varied across $\{100, 200, 300, 400\}$, resulting in a total of 16 combinations. The testing data size of minor loops is taken twice as the FORC in each of the

Table 3
The generalization performance for $B_{max} = 1.25$ T on training data size (N) of 300 and varying testing data size (M) of FORC in the range 100, 200, 300, and 400. The testing data size of minor loops is twice as FORC test size.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	100	200	300	400	100	200	300	400	100	200	300	400	100	200	300	400
$\mathcal{C}_{\text{FORC}_1}$	0.0105	0.0154	0.0216	0.0258	0.9895	0.9879	0.9861	0.9849	0.2386	0.2378	0.2396	0.2398	0.0504	0.0752	0.0940	0.1042
$\mathcal{C}_{\text{FORC}_2}$	0.0628	0.0179	0.0378	0.0455	0.9479	0.9932	0.9915	0.9906	0.3222	0.2537	0.2537	0.2551	0.1515	0.0869	0.1343	0.1480
$\mathcal{C}_{\text{minor}_1}$	0.0498	0.0191	0.0120	0.0113	0.9679	0.9863	0.9887	0.9888	0.2938	0.1373	0.1143	0.1189	0.1036	0.0696	0.0550	0.0532
$\mathcal{C}_{\text{minor}_2}$	0.3646	0.0997	0.0647	0.0636	0.8253	0.9267	0.9366	0.9364	0.5738	0.2513	0.1939	0.2034	0.2296	0.1278	0.1044	0.1035

Table 4
The generalization performance for $B_{max} = 1.25$ T on training data size (N) of 600 and varying testing data size (M) of FORC in the range 100, 200, 300, and 400. The testing data size of minor loops is twice as FORC test size.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	100	200	300	400	100	200	300	400	100	200	300	400	100	200	300	400
$\mathcal{C}_{\text{FORC}_1}$	0.0168	0.0210	0.0211	0.0275	0.9876	0.9862	0.9861	0.9846	0.2387	0.2399	0.2392	0.2402	0.0795	0.0920	0.0923	0.1075
$\mathcal{C}_{\text{FORC}_2}$	0.0218	0.0320	0.0322	0.0392	0.9926	0.9922	0.9921	0.9908	0.2543	0.2540	0.2539	0.2550	0.0978	0.1229	0.1229	0.1359
$\mathcal{C}_{\text{minor}_1}$	0.0163	0.0128	0.0126	0.0114	0.9875	0.9883	0.9885	0.9886	0.1173	0.1147	0.1109	0.1255	0.0644	0.0569	0.0567	0.0535
$\mathcal{C}_{\text{minor}_2}$	0.0838	0.0707	0.0698	0.0671	0.9332	0.9344	0.9344	0.9338	0.2174	0.1905	0.1948	0.2054	0.1175	0.1090	0.1082	0.1062

Table 5
The generalization performance for $B_{max} = 1.25$ T on training data size (N) of 900 and varying testing data size (M) of FORC in the range 100, 200, 300, and 400. The testing data size of minor loops is twice as FORC test size.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	100	200	300	400	100	200	300	400	100	200	300	400	100	200	300	400
$\mathcal{C}_{\text{FORC}_1}$	0.0515	0.0186	0.0149	0.0160	0.9752	0.9869	0.9882	0.9882	0.2838	0.2394	0.2387	0.2389	0.1459	0.0848	0.0721	0.0758
$\mathcal{C}_{\text{FORC}_2}$	0.0173	0.0178	0.0111	0.0128	0.9914	0.9928	0.9898	0.9908	0.2563	0.2544	0.2547	0.2546	0.0793	0.0856	0.0526	0.0578
$\mathcal{C}_{\text{minor}_1}$	0.0125	0.0165	0.0232	0.0211	0.9879	0.9866	0.9834	0.9839	0.1342	0.1263	0.1676	0.1578	0.0545	0.0642	0.0759	0.0727
$\mathcal{C}_{\text{minor}_2}$	0.1122	0.1005	0.1571	0.1386	0.9143	0.9253	0.9049	0.9084	0.2766	0.2585	0.3434	0.3201	0.1352	0.1272	0.1582	0.1499

Table 6
The generalization performance for $B_{max} = 1.25$ T on training data size (N) of 1200 and varying testing data size (M) of FORC in the range 100, 200, 300, and 400. The testing data size of minor loops is twice as FORC test size.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	100	200	300	400	100	200	300	400	100	200	300	400	100	200	300	400
$\mathcal{C}_{\text{FORC}_1}$	0.0213	0.0198	0.0116	0.0207	0.9861	0.9865	0.9890	0.9866	0.2403	0.2391	0.2398	0.2394	0.0932	0.0887	0.0599	0.0902
$\mathcal{C}_{\text{FORC}_2}$	0.0362	0.0255	0.0161	0.0136	0.9916	0.9929	0.9933	0.9916	0.2547	0.2541	0.2553	0.2540	0.1311	0.1084	0.0811	0.0644
$\mathcal{C}_{\text{minor}_1}$	0.0122	0.0143	0.0274	0.0168	0.9886	0.9877	0.9863	0.9856	0.1137	0.1106	0.1622	0.1266	0.0555	0.0605	0.0859	0.0655
$\mathcal{C}_{\text{minor}_2}$	0.0656	0.0812	0.0983	0.1256	0.9363	0.9309	0.9351	0.9124	0.1921	0.2092	0.2450	0.2975	0.1051	0.1166	0.1266	0.1430

16 cases. The results for these cases are presented in Tables 3 – 6. The results indicate that the performance of HystRNN remains consistent across different data configurations, with only minor deviations. The model successfully generalizes the magnetic hysteresis pattern across all tested scenarios.

5.4.2. Comparison with Duhem model of hysteresis

This section compares the performance of HystRNN with the Duhem model. Specifically, the Duhem model (2) is employed to model the major loop, and the learned model is subsequently used to predict FORCs and minor loops—analogous to the approach used for HystRNN.

Implementing the Duhem model involves preparing the required feature terms and enabling the estimation of the model coefficients through regression techniques. Concretely, the objective is to determine the coefficients α, β, and γ in (2) given the feature set $|H'|H, |H'|B$, and H', with the target label B'. These coefficients are estimated using standard least-squares regression and Lasso regression [65], implemented via the `scikit-learn` library [66]. The intercept term is excluded from training to maintain consistency with the form of the Duhem model, which does not include a constant coefficient. The learned coefficients for least-squares regression are $\alpha = 0.402$, $\beta = 0.087$, and $\gamma = 0.325$, while for Lasso regression, they are $\alpha = 0.379$, $\beta = 0.069$, and $\gamma = 0.326$.

Once obtained in this explicit form, the Duhem model is used to predict B fields in the generalization task by numerically solving the resulting implicit ODE using Euler's method. The results of this experiment, presented in Table 7, demonstrate the Duhem model's inability to generalize to novel excitation patterns, particularly FORCs and minor loops when trained on a single major loop. Specifically for both the regression methods, Table 7 presents the poor generalization performance across different test cases (up to two orders of inferior performance than HystRNN in L2-norm, as presented in Table 2). The results show that least-square and Lasso regression exhibit inferior accuracy compared to HystRNN, with least-square regression achieving slightly lower L2-norm and MAE than Lasso. This highlights the generalization capability of the proposed HystRNN model compared to regression methods applied to the Duhem model.

5.4.3. Comparison with vanilla ODE model

This section compares HystRNN to its ODE formulation to assess whether generalization to minor loops and FORCs can be achieved solely by fitting the vanilla version of the proposed ODE model. Specifically, the vanilla linear second-order ODE formulation corresponding to (1) could be written as,

Table 7
The generalization performance for $B_{max} = 1.25$ T using least-square regression and Lasso regression for the Duhem model. EVS and MAE refer to explained variance score and mean absolute error, respectively.

Test case	Least-square regression				Lasso regression			
	L2-norm (↓)	EVS (↑)	Max error (↓)	MAE (↓)	L2-norm (↓)	EVS (↑)	Max error (↓)	MAE (↓)
C_{FORC_1}	2.2478	0.2945	1.5241	0.7967	2.2794	0.2875	1.5363	0.8029
C_{FORC_2}	2.1048	0.3302	1.3804	0.7529	2.1298	0.3235	1.3900	0.7576
C_{minor_1}	2.3261	0.2003	1.5187	0.7234	2.3857	0.2003	1.5300	0.7344
C_{minor_2}	2.6041	0.1834	1.2902	0.6202	2.6518	0.1825	1.2962	0.6274

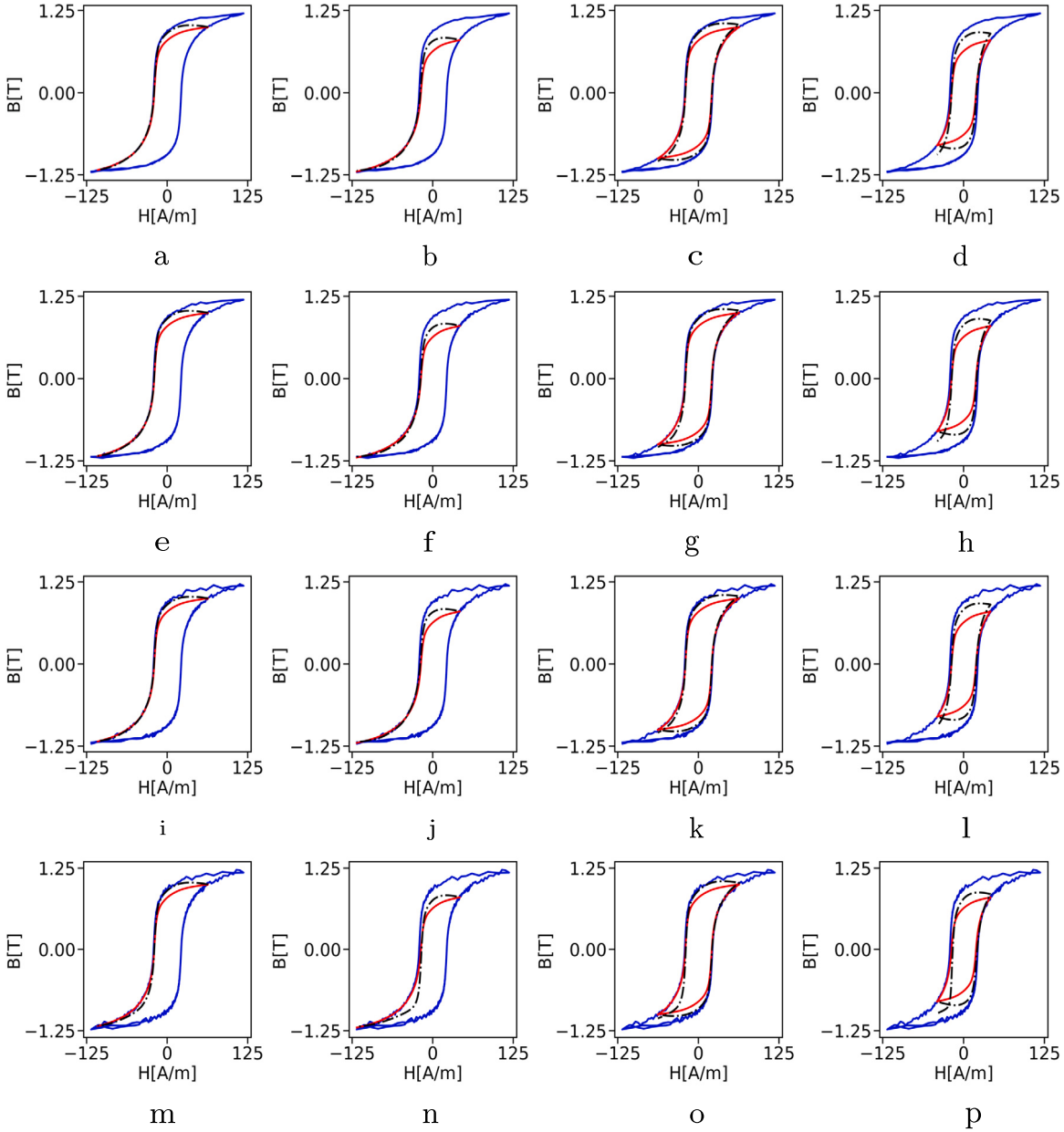


Fig. 6. Experimental vs predicted hysteresis trajectories in case of noisy training data for HystRNN, where $B_{max} = 1.25$ T. Rows represent solutions for different noise levels, 0.5% Gaussian noise (6a, 6b, 6c, 6d), 1% Gaussian noise (6e, 6f, 6g, 6h), 1.5% Gaussian noise (6i, 6j, 6k, 6l), 2% Gaussian noise (6m, 6n, 6o, 6p). Columns represent predictions for C_{FORC_1}, C_{FORC_2}, C_{minor_1}, and C_{minor_2} from left to right.

$$B'' = \alpha_1 B + \alpha_2 B' + \alpha_3 H + \alpha_4 |B|^2 + \alpha_5 |B'|^2 + \alpha_6 |H|^2 + \alpha_7, \quad (5)$$

where α_1, …, α_7 are learnable parameters. As in the previous section, these parameters are estimated through the least-squares and Lasso regression by fitting the ODE model for major loop data. The trained model is then evaluated on FORCs and minor loops to assess its generalization capability. The results, presented in Table 8, demonstrate that this approach fails to generalize effectively, performing even worse than the Duhem model fitted via regression and exhibiting higher errors and significantly negative explained variance scores, indicating poor dataset generalization.

Table 8
The generalization performance for $B_{max} = 1.25$ T using least-square regression and lasso regression for the vanilla ODE. EVS and MAE refer to explained variance score and mean absolute error, respectively.

Test case	Least-square regression				Lasso regression			
	L2-norm (↓)	EVS (↑)	Max error (↓)	MAE (↓)	L2-norm (↓)	EVS (↑)	Max error (↓)	MAE (↓)
$\mathcal{C}_{\text{FORC}_1}$	8557.0675	-8654.4295	119.9668	33.9983	9064.8328	-9196.6606	119.9668	35.4125
$\mathcal{C}_{\text{FORC}_2}$	7.6944	-7.6641	21.9505	0.7179	10.2647	-10.5528	25.6381	0.7504
$\mathcal{C}_{\text{minor}_1}$	207.5543	-93.3281	15.3325	6.1792	206.8274	-90.5530	18.5410	6.2922
$\mathcal{C}_{\text{minor}_2}$	317.3377	-147.3552	15.1147	6.0583	286.9747	-136.9473	13.9428	5.7018

Table 9
The generalization performance for $B_{max} = 1.25$ T on varying ODE step sizes of 0.005, 0.01, 0.02, and 0.1.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	0.005	0.01	0.02	0.1	0.005	0.01	0.02	0.1	0.005	0.01	0.02	0.1	0.005	0.01	0.02	0.1
$\mathcal{C}_{\text{FORC}_1}$	1.0096	0.0109	0.0276	0.0201	-0.0272	0.9891	0.9745	0.9865	1.3509	0.2370	0.4207	0.2386	0.6672	0.0563	0.0903	0.0890
$\mathcal{C}_{\text{FORC}_2}$	0.9949	0.0115	0.0457	0.0219	-0.0611	0.9924	0.9828	0.9915	1.3774	0.2520	0.4336	0.2587	0.6316	0.0583	0.0936	0.0962
$\mathcal{C}_{\text{minor}_1}$	0.8637	0.0320	0.0187	0.0159	0.1388	0.9844	0.9900	0.9876	1.0587	0.1882	0.1456	0.1506	0.4543	0.0916	0.0672	0.1062
$\mathcal{C}_{\text{minor}_2}$	0.8597	0.1283	0.0691	0.0944	0.1514	0.9267	0.9340	0.9290	0.9018	0.2923	0.2091	0.2729	0.3601	0.1443	0.1067	0.1195

Table 10
The generalization performance for $B_{max} = 1.25$ T on DC-bias data.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN
$\mathcal{C}_{\text{FORC}_1}$	4.9107	1.2323	0.6664	0.0207	0.9815	0.9182	0.9097	0.9863	2.5139	1.4930	1.0782	0.2394	1.5539	0.6875	0.5377	0.0910
$\mathcal{C}_{\text{FORC}_2}$	5.9152	0.9582	0.4960	0.0261	0.9700	0.9180	0.8895	0.9928	2.5086	1.2993	0.8844	0.2540	1.6755	0.5801	0.4546	0.1095
$\mathcal{C}_{\text{minor}_1}$	6.9548	1.6321	0.7814	0.0136	0.3135	0.8914	0.8662	0.9879	2.2892	1.3353	0.9247	0.1128	1.4292	0.5980	0.4347	0.0590
$\mathcal{C}_{\text{minor}_2}$	10.8439	2.1834	0.8483	0.0803	0.2135	0.8873	0.8606	0.9302	2.1878	1.1794	0.7717	0.2040	1.4551	0.5647	0.3596	0.1160

The results suggest that the ability to generalize hysteresis is not solely attributed to the ODE formulation but rather to the interplay between the recurrent architecture and the hidden state updates in HystRNN. The recurrent structure allows the model to learn complex hysteresis patterns dynamically, enabling robust generalization to previously unseen FORCs and minor loops. In contrast, even when optimized using regression, the ODE-based approach lacks the adaptability required for such generalization.

5.4.4. Ablation on the ODE step size

This Section discusses the impact of ODE step size and the computational cost of HystRNN. An experiment is performed by varying Δt across the values $\{0.005, 0.01, 0.02, 0.1\}$ to evaluate the impact of Δt on HystRNN's performance. The results in Table 9 demonstrate that the method remains robust across different step sizes, achieving a close approximation in generalization tasks and consistently outperforming RNN, LSTM, and GRU. These results indicate that the choice of Δt does not significantly impact the smoothness of the solution. However, there is no universal trend for selecting Δt, as the optimal value depends on the specific problem and the prediction target—whether FORCs or minor loops.

Additionally, HystRNN requires 8.86 ± 0.22 seconds on average to train, whereas GRU takes 640.82 ± 11.15 seconds for the same number of epochs. The observed run time demonstrates that HystRNN in addition of being accurate is also more computationally efficient than GRU. The computational time is averaged over five runs, with the training time recorded in each run. The reported mean and standard deviation further validate the efficiency of HystRNN.

5.4.5. Noisy data

This section presents numerical experiments evaluating the robustness of the proposed method in the presence of noisy training data. Noisy or corrupted magnetic measurement data may present a challenge in modeling magnetic materials. In practice, noise can arise from diverse sources, such as uncontrolled experimental conditions, inherent limitations of measurement devices, and external interferences. A modeling approach invariant to a certain amount of noise is more suited to capture real-world scenarios, enhancing the applicability of the developed model. By subjecting HystRNN to rigorous noise assessment, the experiments present its capacity to maintain accuracy amidst noisy data collection scenarios, enhancing its reliability and applicability in the characterization of magnetic materials under realistic conditions.

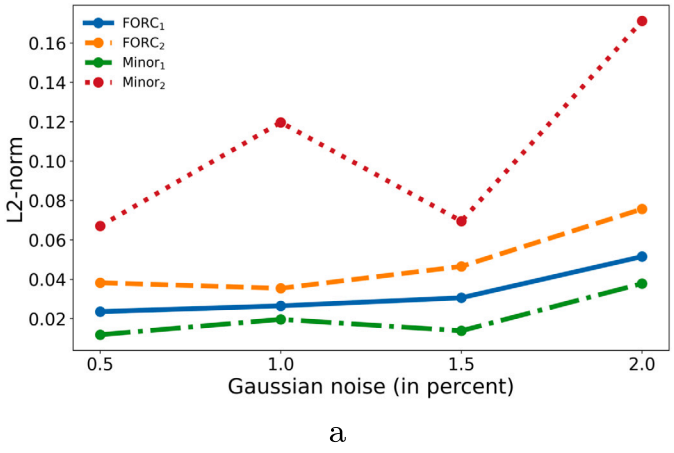


Fig. 7. L2-norm varying with Gaussian noise: HystRNN for the second experiment.

In particular, the training data for experiment 2, which is for $B_{max} = 1.25$ T, is subjected to noise before training HystRNN. Experiments for other recurrent models are not performed for noisy training data as they could not be generalized even for noise-free data. For HystRNN, the data measured for the major loop is corrupted with four distinct noise levels. Specifically, Gaussian noise with 0.5, 1, 1.5 and 2 percent is applied. Subsequently, FORCs and minor loops are tested similarly to those in the previous experiments. The origin of $\mathcal{C}_{\text{FORC}_{1,2}}$ is kept same as in experiment 2, to be 1.2 T and 1.0 T respectively. The chosen maximum B

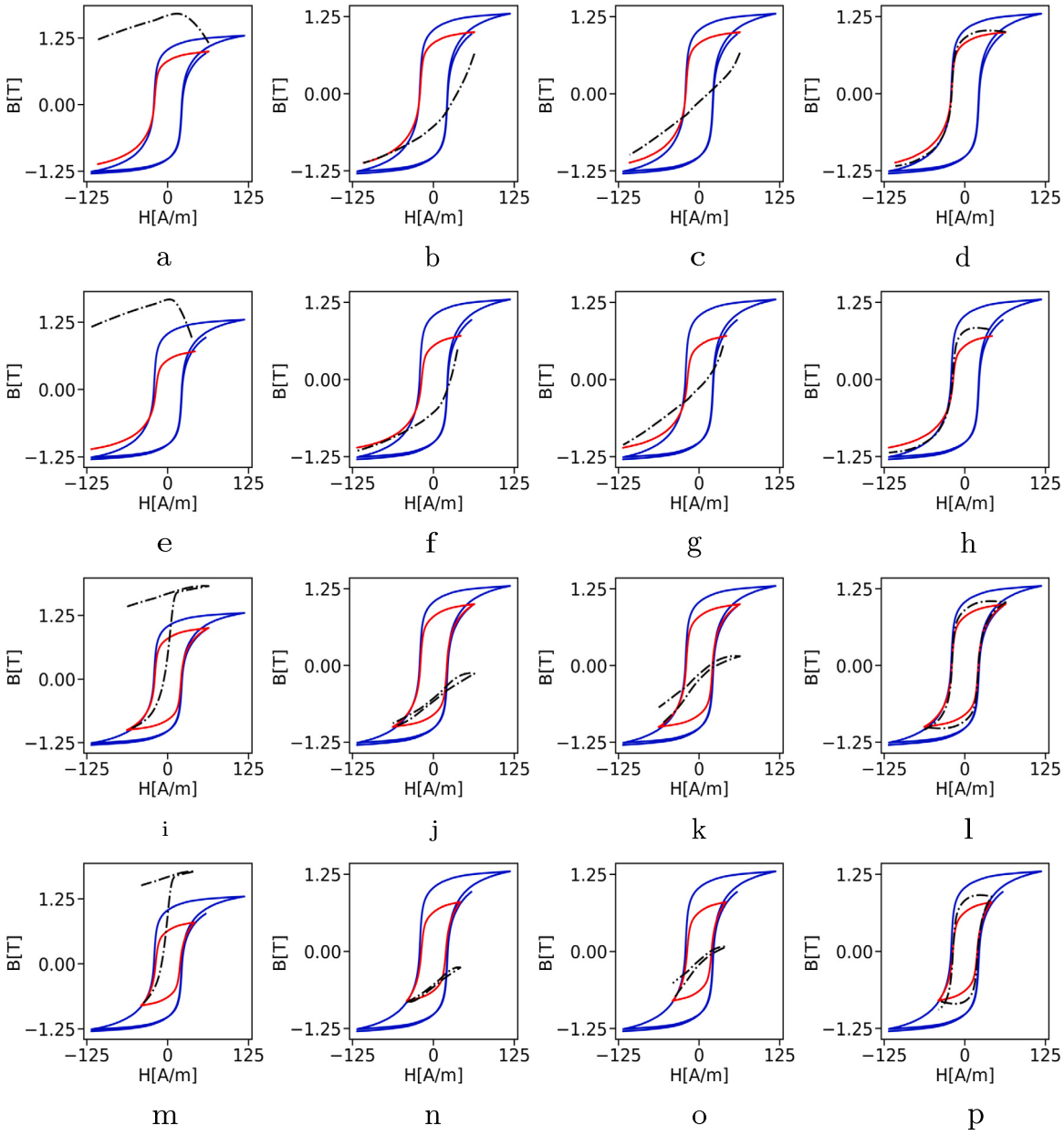


Fig. 8. Experimental vs predicted hysteresis trajectories for DC-bias experiment, where $B_{max} = 1.25$ T. Columns represent solutions for different methods, RNN(8a, 8e, 8i, 8m), LSTM(8b, 8f, 8j, 8n), GRU(8c, 8g, 8k, 8o), HystRNN(8d, 8h, 8l, 8p). Top two rows: predictions for $\mathcal{C}_{\text{FORC}_1}$ and $\mathcal{C}_{\text{FORC}_2}$ respectively. Bottom two rows: predictions for $\mathcal{C}_{\text{minor}_1}$ and $\mathcal{C}_{\text{minor}_2}$ respectively.

value of $\mathcal{C}_{\text{minor}_{1,2}}$ is 1.0 T and 0.8 T. The experiment results are presented in Fig. 6 and 7.

Fig. 6 presents the performance of HystRNN in case of noisy training data. Specifically, the first two columns portray the prediction for the first-order reversal curves, while the latter two depict minor loop predictions. The top two rows of Fig. 6 exhibit noise levels of 0.5 and 1 percent, while the bottom two rows show 1.5 and 2 percent noise levels, respectively. Despite increasing the noise levels in the training data, a slight discrepancy is observed in prediction accuracy compared to the noise-free conditions outlined in Table 2. Notably, the accuracy remains consistent under an order of magnitude change. Additionally, no post-processing or smoothing procedure has been applied to the predicted curve. The predictions presented in Fig. 6 are directly obtained from the proposed approach, HystRNN, without any additional modifications. The smooth appearance of the predicted curve is an inherent result of the model's ability to accurately capture the underlying hysteresis dynamics while regularizing high-frequency noise in the data.

Fig. 7 illustrates the L2-norm for all four predictions for first-order reversal curves and minor loops across different noise levels $(0.5\%, 1\%, 1.5\%, 2\%)$ for HystRNN. There is a consistent trend of increasing error at the highest noise level in all cases. Notably, the errors for the first-order reversal curves do not spike significantly with increased noise, highlighting the method's robustness.

5.4.6. *Performance on DC-bias*

In real-world scenarios, excitation signals often contain DC-bias, making it essential to assess whether the proposed HystRNN model can generalize effectively under such conditions. This section presents an additional experiment to evaluate the robustness of HystRNN against these perturbations.

To introduce DC-bias into the testing data, a bias function is added to the excitation signal given by $0.1\sin(2\pi t) + 0.001 \cdot \mathcal{N}(0,1)$, where $\mathcal{N}(0,1)$ represents Gaussian noise with zero mean and unit variance

and t represents the data sequence step. The modified excitation is then used to test the predictive performance of all trained models.

The results, summarized in Fig. 8 and Table 10, indicate that HystRNN maintains high predictive accuracy despite the introduction of DC-bias. The table compares the generalization performance of different machine learning models (RNN, LSTM, GRU, and HystRNN) on DC-bias data with max(B) = 1.25 T using four metrics. HystRNN consistently outperforms other models, achieving the lowest L2-norm, max, and mean absolute error while maintaining the highest explained variance score. The reported errors remain nearly unchanged, demonstrating the robustness of the model. In contrast, conventional recurrent architectures—RNN, LSTM, and GRU—suffer from a similar low accuracy, reinforcing previous findings that these methods struggle with generalization when trained on limited data.

Overall, these experiments highlight that HystRNN remains resilient in the presence of DC-bias and retains its capability to generalize complex hysteresis patterns that inherently include high-order harmonics. These findings further emphasize the superiority of HystRNN over conventional recurrent architectures in modeling and predicting hysteresis behavior under realistic conditions.

5.5. Experiment 3

For the third experiment, $B_{max} = 1.3$ T. The origin of $C_{FORC_{1,2}}$ is taken to be 1.0 T and 0.75 T respectively. The chosen maximum B value of $C_{minor_{1,2}}$ is 1.0 T and 0.75 T. The results (Fig. D.1) and numerical comparisons in tabular form (Table D.1) are presented in Appendix D and validate the superiority of the proposed method compared to the traditional recurrent networks.

5.6. Experiment 4

For the fourth experiment, $B_{max} = 1.5$ T. The origin of $C_{FORC_{1,2}}$ is taken to be 1.25 T and 0.75 T respectively. The chosen maximum B value of $C_{minor_{1,2}}$ is 0.9 T and 0.7 T. The results (Fig. D.2) and numerical comparisons in tabular form (Table D.2) are supplement in Appendix D for conciseness.

6. Conclusions

This paper introduced a physics-aware recurrent neural network, HystRNN, aimed to advance magnetic hysteresis modeling within extrapolated regions. The proposed method is inspired from phenomenological hysteresis models for hidden state update of recurrent neural networks. HystRNN was validated by predicting first-order reversal curves and minor loops after training the model solely with major loop data. The outcomes underscore the superiority of HystRNN in adeptly capturing intricate nonlinear dynamics, outperforming conventional recurrent neural architectures such as RNN, LSTM, and GRU on various metrics. The performance of HystRNN is further validated by its performance on noisy training data, showcasing robustness for upto 2% Gaussian noise. This performance is attributed to its capacity to assimilate sequential information, history dependencies, and hysteretic features, ultimately achieving generalization capabilities.

CRediT authorship contribution statement

Abhishek Chandra: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Taniya Kapoor:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Bram Daniels:** Writing – original draft, Software, Data curation. **Mitrofan Curti:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Koen Tiels:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Daniel M. Tartakovsky:** Writing – review & editing, Supervision. **Elena A. Lomonova:** Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Data generation through Preisach model of hysteresis

Preisach model of hysteresis is used to generate the data used for this work. It is a phenomenological model that describes the hysteresis effect by a set of hysteresis operators, scattered on a triangular domain called the Preisach plane, which are scaled by a weight function, following

$$f(t) = \iint_{\alpha \geq \beta} P(\alpha, \beta) \hat{\gamma}_{\alpha,\beta} w(t) d\alpha d\beta \tag{6}$$

where w is the input and f is the output. The moment of time is represented by t and α& β are two switching variables. P is the Preisach weight function, and $\hat{\gamma}$ represents the hysteresis operators.

In this work the static scalar Preisach model is used with a weight function fitted on a set of measurement data. The data are measured by a Brockhaus MPG 200 soft-magnetic material tester, using an Epstein frame calibrated to correspond with the IEC standard as shown in Fig. A.1. A set of concentric hysteresis loops up to a maximum of 1.7 T is measured for NO27-1450H, obtained under quasi-DC excitation. Here, quasi-DC indicates that the rate of change of the magnetic flux density is controlled such that any eddy current fields are negligible, and the static hysteresis behavior is obtained. The Epstein sample strips used are obtained using spark eroding and cut in the rolling direction.



Fig. A.1. The MPG 200 desk by Brockhaus, with Epstein frame and sample strips shown on the right.

Appendix B. Data normalization

This work utilized min-max scaling to transform measured data to fall within a specific range, between -1 and 1. The purpose of this normalization is to bring all the features to a similar scale, important for recurrent architectures, as their training is sensitive to the scale of input data. The min-max scaling is defined as,

$$\mathbf{u}_{scaled} = \frac{\mathbf{u} - \mathbf{u}_{min}}{\mathbf{u}_{max} - \mathbf{u}_{min}} \cdot 2 - 1$$

where $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ represent the minimum and maximum value in the original dataset $\mathbf{u}$, respectively. Dividing by $(\mathbf{u}_{\max} - \mathbf{u}_{\min})$ scales the data to the range $[0, 1]$, and multiplying by 2 and subtracting 1 scales the data to the range $[-1, 1]$. The following expression is used to revert the scaled data back to its original values

$$\mathbf{u}_{\text{original}} = \frac{\mathbf{u}_{\text{scaled}} + 1}{2} \cdot (\mathbf{u}_{\max} - \mathbf{u}_{\min}) + \mathbf{u}_{\min}$$

Appendix C. Error metrics

The error metrics used in this study are described as follows,

$L2$-norm: The relative $L2$-norm of $\hat{\mathrm{B}}$ with respect to ground truth B is $\frac{\|\hat{\mathrm{B}}-\mathrm{B}\|_2}{\|\mathrm{B}\|_2}$. Here, $\|\hat{\mathrm{B}} - \mathrm{B}\|_2$ is the Euclidean distance between $\hat{\mathrm{B}}$ and B, and $\|\mathrm{B}\|_2$ is the Euclidean norm of B.

Explained variance score: The second metric, explained variance score is given by

$$1 - \frac{\sum_{i=1}^{n}(\mathrm{B}_i - \hat{\mathrm{B}}_i)^2}{\sum_{i=1}^{n}(\mathrm{B}_i - \bar{\mathrm{B}})^2}$$

where n is the number of data points, B_i represents the ground truth at the i^{th} data point, $\hat{\mathrm{B}}_i$ represents the predicted value at the i^{th} data point, and $\bar{\mathrm{B}}$ represents the mean of the ground truth.

Max error: The third metric is the maximum absolute error computed by, $\max_{i=1}^{n} |\mathrm{B}_i - \hat{\mathrm{B}}_i|$. Here, $|.|$ represents the absolute value function.

Mean absolute error: The fourth metric, mean absolute error is given by, $\frac{1}{n}\sum_{i=1}^{n} |\mathrm{B}_i - \hat{\mathrm{B}}_i|$, where the symbols have their usual meaning.

Appendix D. Results for experiment 3 and 4

For experiment 3, the predictions for the model, the ground truth, and the training data are presented in Fig. D.1. As presented in the first two rows by Fig. D.1a, D.1e, D.1b, D.1f, D.1c, and D.1g predictions by RNN, LSTM and GRU models exhibit a lack of accuracy. In contrast,

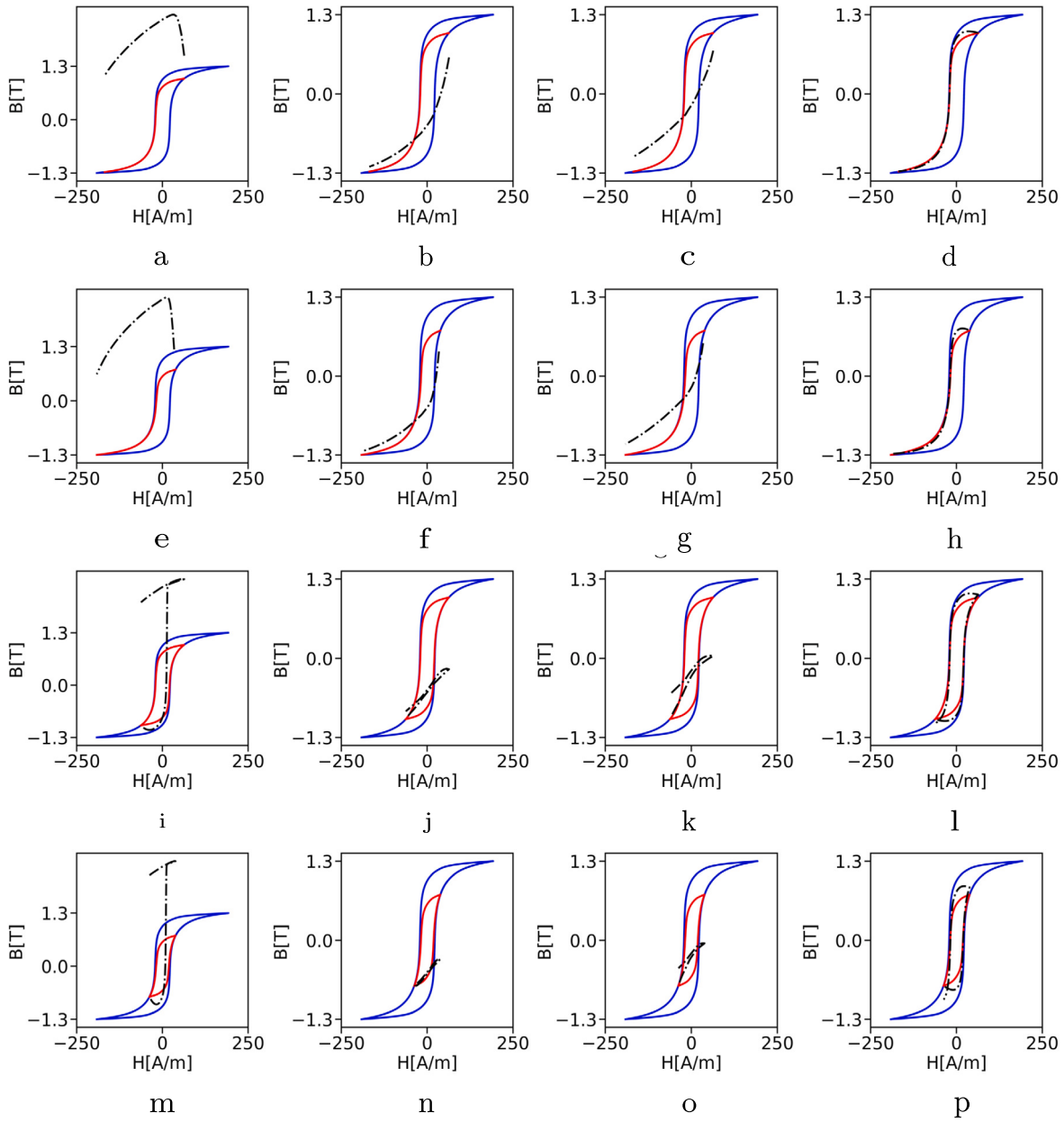


Fig. D.1. Experimental vs predicted hysteresis trajectories for experiment 3, where $\mathrm{B}_{\max} = 1.3\,\mathrm{T}$. Columns represent solutions for different methods, RNN(D.1a, D.1e, D.1i, D.1m), LSTM (D.1b, D.1f, D.1j, D.1n), GRU(D.1c, D.1g, D.1k, D.1o), HystRNN(D.1d, D.1h, D.1l, D.1p). Top two rows: predictions for C_{FORC_1} and C_{FORC_2} respectively. Bottom two rows: predictions for C_{minor_1} and C_{minor_2} respectively.

Table D.1
The generalization performance assessed using the metrics: L2-norm relative error, explained variance error, maximum error, and mean absolute error for experiment 3, where $B_{max} = 1.3$ T.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN
$\mathcal{C}_{\text{FORC}_1}$	6.0907	0.9692	0.6625	**0.0342**	0.0534	0.3290	0.3649	**0.9705**	3.2184	1.5661	1.2194	**0.3878**	1.9992	0.7105	0.6247	**0.1296**
$\mathcal{C}_{\text{FORC}_2}$	7.2661	0.6720	0.4775	**0.0377**	0.0295	0.3691	0.4183	**0.9785**	3.2126	1.2972	0.9456	**0.4055**	2.1660	0.5806	0.5249	**0.1371**
$\mathcal{C}_{\text{minor}_1}$	10.2305	1.6042	0.9009	**0.0301**	-0.0216	0.2699	0.2909	**0.9774**	2.7703	1.3222	0.9947	**0.1857**	1.7520	0.5923	0.4619	**0.0855**
$\mathcal{C}_{\text{minor}_2}$	18.2528	2.3069	1.0498	**0.1580**	-0.1629	0.2528	0.2785	**0.8780**	2.5696	1.1249	0.8045	**0.3066**	1.7901	0.5446	0.3673	**0.1491**

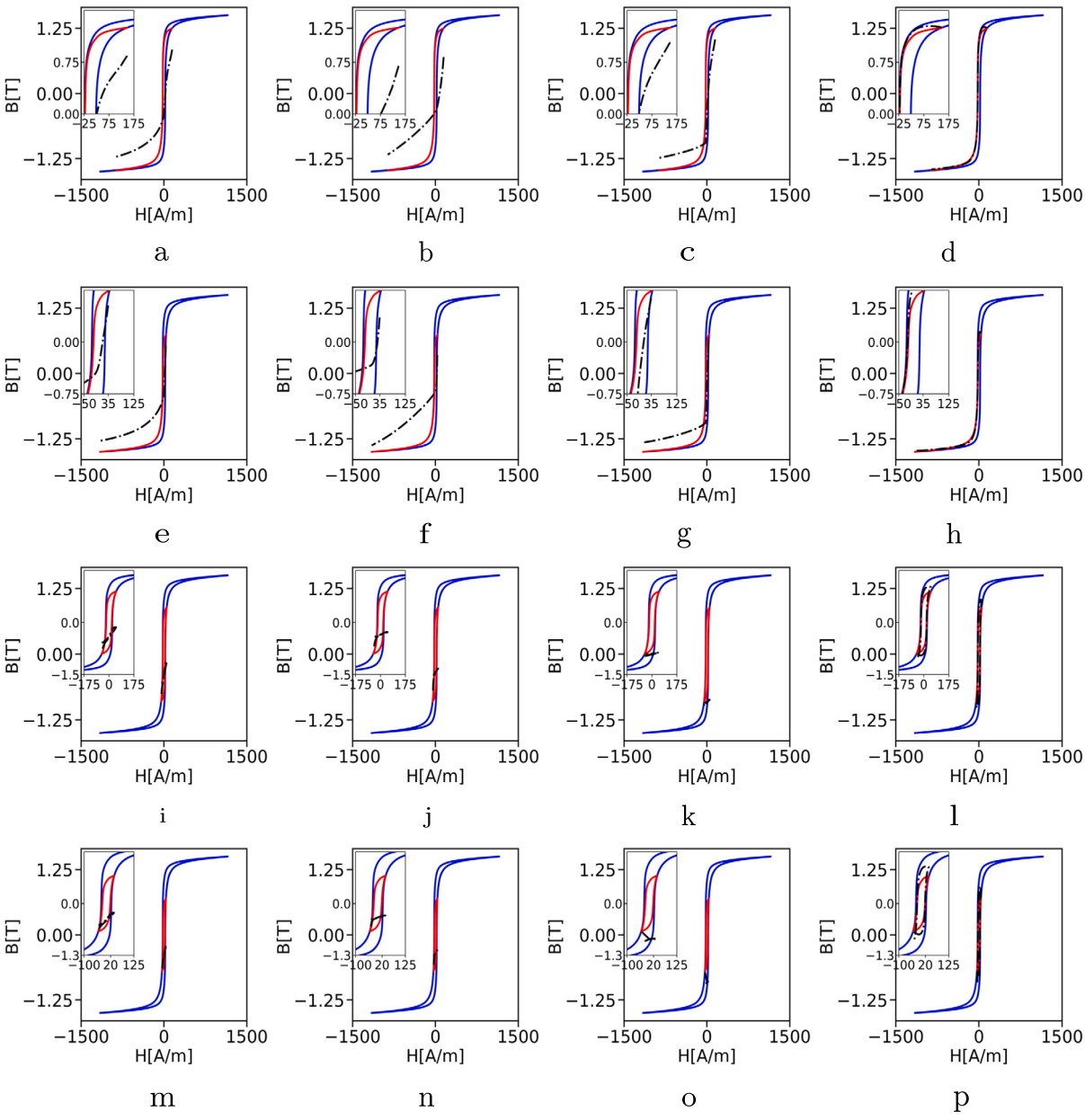


Fig. D.2. Experimental vs predicted hysteresis trajectories for experiment 4, where $B_{max} = 1.5$ T. Columns represent solutions for different methods, RNN (D.2a, D.2e, D.2i, D.2m), LSTM (D.2b, D.2f, D.2j, D.2n), GRU (D.2c, D.2g, D.2k, D.2o), HystRNN (D.2d, D.2h, D.2l, D.2p). Top two rows: predictions for $\mathcal{C}_{\text{FORpredictedC}_1}$ and $\mathcal{C}_{\text{FORC}_2}$ respectively. Bottom two rows: predictions for $\mathcal{C}_{\text{minor}_1}$ and $\mathcal{C}_{\text{minor}_2}$ respectively.

predictions of HystRNN for the reversal curve are notably precise, as evidenced in Fig. D.1d and D.1h. The final two rows within Fig. D.1 present that HystRNN accurately captured the characteristics of the minor loop, as showcased in Fig. D.1l and D.1p. Prediction by GRU manages to capture a resemblance of the loop, although not entirely, as revealed in Fig. D.1k and D.1o. On the other hand, LSTM and RNN perform poorly, failing to capture the intricate structure of the minor loop, as depicted in Fig. D.1j, D.1n, D.1i, and D.1m respectively.

For experiment 4, the predictions for the model, the ground truth, and the training data are presented in Fig. D.2. Predictions of the reversal curve show agreement with the nature observed in previous experiments. In this case, max(B), origin of $\mathcal{C}_{\text{FORC}_2}$, and maximum B

Table D.2
The generalization performance assessed using the metrics: L2-norm relative error, explained variance error, maximum error, and mean absolute error for experiment 4, where $B_{max} = 1.5$ T.

Test case	L2-norm (↓)				Explained variance score (↑)				Max error (↓)				Mean absolute error (↓)			
	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN	RNN	LSTM	GRU	HystRNN
C_{FORC_1}	0.7673	0.8556	0.9330	**0.1007**	0.2957	0.1675	0.3407	**0.9017**	1.7979	1.9572	1.8971	**0.7491**	0.9147	0.9642	0.9957	**0.3258**
C_{FORC_2}	0.4968	0.5942	0.5268	**0.1164**	0.3331	0.2209	0.3718	**0.9330**	1.2239	1.3492	1.3040	**0.7725**	0.7094	0.7686	0.7253	**0.3304**
C_{minor_1}	1.3719	1.3723	4.0550	**0.0784**	0.2318	0.0903	0.0640	**0.9224**	1.1112	1.1767	1.7688	**0.2394**	0.4981	0.5022	0.9177	**0.1306**
C_{minor_2}	1.7923	1.6721	6.0749	**0.2257**	0.2291	0.0907	0.0157	**0.7743**	0.9550	0.9988	1.5820	**0.2991**	0.4396	0.4261	0.911	**0.1717**

value of C_{minor_2} vary significantly, posing a challenge for both LSTM and GRU. However, HystRNN outperforms them for each case, as shown in Fig. D.2. The results underscore the performance of HystRNN as, for neither of the cases, the accuracy of LSTM or GRU is comparable to the proposed method.

Data availability

I have shared the link to data and code in the revised manuscript

References

[1] G. Bertotti, I.D. Mayergoyz, The Science of Hysteresis: 3-Volume Set, Elsevier, 2005.
[2] D. Wilms, A. Winkler, P. Virnau, K. Binder, Monte Carlo simulations of the 2d-Ising model in the geometry of a long stripe, Comput. Phys. Commun. 182 (9) (2011) 1892–1895.
[3] R.P. Dhote, H. Gómez, R.N. Melnik, J. Zu, Shape memory alloy nanostructures with coupled dynamic thermo-mechanical effects, Comput. Phys. Commun. 192 (2015) 48–53.
[4] L. Exl, J. Fischbacher, A. Kovacs, H. Oezelt, M. Gusenbauer, T. Schrefl, Preconditioned nonlinear conjugate gradient method for micromagnetic energy minimization, Comput. Phys. Commun. 235 (2019) 179–186.
[5] S. Quondam-Antonio, F. Riganti-Fulginei, A. Laudani, G.-M. Lozito, R. Scorretti, Deep neural networks for the efficient simulation of macro-scale hysteresis processes with generic excitation waveforms, Eng. Appl. Artif. Intell. 121 (2023) 105940.
[6] S. Krim, M.F. Mimouni, Design of improved direct torque control based on a five level torque controller and a new Sugeno–Takagi fuzzy super-twisting controller applied to an induction machine, Eng. Appl. Artif. Intell. 126 (2023) 106900.
[7] D. Ceylan, R. Zeinali, B. Daniels, K.O. Boynov, E.A. Lomonova, A novel modeling technique via coupled magnetic equivalent circuit with vector hysteresis characteristics of laminated steels, IEEE Trans. Ind. Appl. 59 (2) (2022) 1481–1491.
[8] G. Chen, G. Chen, Y. Lou, Diagonal recurrent neural network-based hysteresis modeling, IEEE Trans. Neural Netw. Learn. Syst. 33 (12) (2021) 7502–7512.
[9] G. Bertotti, Hysteresis in Magnetism: for Physicists, Materials Scientists, and Engineers, Gulf Professional Publishing, 1998.
[10] F. Preisach, Über die magnetische nachwirkung, Z. Phys. 94 (5–6) (1935) 277–302.
[11] D.C. Jiles, D.L. Atherton, Theory of ferromagnetic hysteresis, J. Magn. Magn. Mater. 61 (1–2) (1986) 48–60.
[12] R. Bouc, Forced vibrations of mechanical systems with hysteresis, in: Proc. of the Fourth Conference on Nonlinear Oscillations, Prague, 1967, 1967.
[13] Y.-K. Wen, Method for random vibration of hysteretic systems, J. Eng. Mech. Div. 102 (2) (1976) 249–263.
[14] A. Chandra, B. Daniels, M. Curti, K. Tiels, E.A. Lomonova, D.M. Tartakovsky, Discovery of sparse hysteresis models for piezoelectric materials, Appl. Phys. Lett. 122 (21) (2023).
[15] M. Lin, C. Cheng, G. Zhang, B. Zhao, Z. Peng, G. Meng, Identification of Bouc–Wen hysteretic systems based on a joint optimization approach, Mech. Syst. Signal Process. 180 (2022) 109404.
[16] C. Serpico, C. Visone, Magnetic hysteresis modeling via feed-forward neural networks, IEEE Trans. Magn. 34 (3) (1998) 623–628.
[17] D. Makaveev, L. Dupré, M. De Wulf, J. Melkebeek, Modeling of quasistatic magnetic hysteresis with feed-forward neural networks, J. Appl. Phys. 89 (11) (2001) 6737–6739.
[18] M. Farrokh, F. Dizaji, M. Dizaji, Hysteresis identification using extended Preisach neural network, Neural Process. Lett. (2022) 1–25.
[19] C.-L. Hwang, C. Jan, Recurrent-neural-network-based multivariable adaptive control for a class of nonlinear dynamic systems with time-varying delay, IEEE Trans. Neural Netw. Learn. Syst. 27 (2) (2015) 388–401.
[20] M. Saghafifar, A. Kundu, A. Nafalski, Dynamic magnetic hysteresis modelling using Elman recurrent neural network, Int. J. Appl. Electromagn. Mech. 13 (1–4) (2002) 209–214.
[21] V. Warke, S. Kumar, A. Bongale, P. Kamat, K. Kotecha, G. Selvachandran, A. Abraham, Improving the useful life of tools using active vibration control through data-driven approaches: a systematic literature review, Eng. Appl. Artif. Intell. 128 (2024) 107367.
[22] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv preprint, arXiv:1406.1078, 2014.
[23] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[24] A. Chandra, B. Daniels, M. Curti, K. Tiels, E.A. Lomonova, Magnetic hysteresis modeling with neural operators, IEEE Trans. Magn. 61 (1) (2025) 7300111.
[25] D. Serrano, H. Li, S. Wang, T. Guillod, M. Luo, V. Bansal, N.K. Jha, Y. Chen, C.R. Sullivan, M. Chen, Why magnet: quantifying the complexity of modeling power magnetic material characteristics, IEEE Trans. Power Electron. (2023).
[26] R.T. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations, Adv. Neural Inf. Process. Syst. 31 (2018).
[27] Y. Rubanova, R.T. Chen, D.K. Duvenaud, Latent ordinary differential equations for irregularly-sampled time series, Adv. Neural Inf. Process. Syst. 32 (2019).
[28] T.K. Rusch, S. Mishra, Coupled oscillatory recurrent neural network (CoRNN): an accurate and (gradient) stable architecture for learning long time dependencies, in: International Conference on Learning Representations, 2021.
[29] T.K. Rusch, S. Mishra, N.B. Erichson, M.W. Mahoney, Long expressive memory for sequence modeling, in: International Conference on Learning Representations, 2022.
[30] A. Queiruga, N.B. Erichson, L. Hodgkinson, M.W. Mahoney, Stateful ODE-nets using basis function expansions, Adv. Neural Inf. Process. Syst. 34 (2021) 21770–21781.
[31] N.N. Son, C. Van Kien, H.P.H. Anh, Parameters identification of Bouc–Wen hysteresis model for piezoelectric actuators using hybrid adaptive differential evolution and Jaya algorithm, Eng. Appl. Artif. Intell. 87 (2020) 103317.
[32] J. Oh, D.S. Bernstein, Semilinear Duhem model for rate-independent and rate-dependent hysteresis, IEEE Trans. Autom. Control 50 (5) (2005) 631–645.
[33] A. Chandra, J. Bakarji, D.M. Tartakovsky, Role of physics in physics-informed machine learning, J. Mach. Learn. Mod. Comput. 5 (1) (2024).
[34] Y. Wang, R. Xu, M. Zhou, Prandtl-Ishlinskii modeling for giant magnetostrictive actuator based on internal time-delay recurrent neural network, IEEE Trans. Magn. 54 (11) (2018) 1–4, https://doi.org/10.1109/TMAG.2018.2827397.
[35] H. Zhang, Q. Yang, C. Zhang, Y. Li, Y. Chen, Magnetic properties simulation of electrical steel sheet based on recurrent neural network, IEEE Trans. Magn. 60 (3) (2024) 1–4, https://doi.org/10.1109/TMAG.2023.3316753.
[36] G. Chen, Y. Lou, Recurrent-neural-network-based rate-dependent hysteresis modeling and feedforward torque control of the magnetorheological clutch, IEEE/ASME Trans. Mechatron. 27 (5) (2022) 2875–2886, https://doi.org/10.1109/TMECH.2021.3121498.
[37] H. Li, D. Serrano, T. Guillod, E. Dogariu, A. Nadler, S. Wang, M. Luo, V. Bansal, Y. Chen, C.R. Sullivan, et al., Magnet: an open-source database for data-driven magnetic core loss modeling, in: 2022 IEEE Applied Power Electronics Conference and Exposition (APEC), IEEE, 2022, pp. 588–595.
[38] H. Li, D. Serrano, T. Guillod, S. Wang, E. Dogariu, A. Nadler, M. Luo, V. Bansal, N.K. Jha, Y. Chen, et al., How magnet: machine learning framework for modeling power magnetic material characteristics, IEEE Trans. Power Electron. (2023).
[39] H. Li, D. Serrano, S. Wang, M. Chen, Magnet-ai: neural network as datasheet for magnetics modeling and material recommendation, IEEE Trans. Power Electron. (2023).
[40] Z. Ke, X. Liu, H. Yi, K. Jiang, L. Wang, Z. Deng, Nonlinear levitation-guidance coupling force prediction for HTS pinning Maglev under arbitrary motion based on gated recurrent unit, IEEE Trans. Appl. Supercond. 34 (3) (2024) 1–6, https://doi.org/10.1109/TASC.2024.3356460.
[41] A. Khan, S. Ceesay, Y.-P. Teng, R. Wang, H. Yue, D. Lowther, Generalizable deep neural network based multi-material hysteresis modeling, in: 2022 IEEE 20th Biennial Conference on Electromagnetic Field Computation-Long Papers (CEFC-LONG), IEEE, 2022, pp. 1–4.
[42] S. Steentjes, K. Hameyer, D. Dolinar, M. Petrun, Iron-loss and magnetic hysteresis under arbitrary waveforms in no electrical steel: a comparative study of hysteresis models, IEEE Trans. Ind. Electron. 64 (3) (2016) 2511–2521.
[43] M. Jung, P.R. da Costa Mendes, M. Önnheim, E. Gustavsson, Model predictive control when utilizing LSTM as dynamic models, Eng. Appl. Artif. Intell. 123 (2023) 106226.
[44] T. Kapoor, H. Wang, A. Stamou, K.E. Sayed, A. Núñez, D.M. Tartakovsky, R. Dollevoet, Neural differential equation-based two-stage approach for generalization of beam dynamics, IEEE Trans. Ind. Inform. (2024) 1–10.
[45] Y. Yu, H. Yao, Y. Liu, Structural dynamics simulation using a novel physics-guided machine learning method, Eng. Appl. Artif. Intell. 96 (2020) 103947.
[46] S. Lanthaler, T.K. Rusch, S. Mishra, Neural oscillators are universal, arXiv preprint, arXiv:2305.08753, 2023.

[47] T. Kapoor, A. Chandra, D.M. Tartakovsky, H. Wang, A. Núñez, R. Dollevoet, Neural oscillators for generalization of physics-informed machine learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, 2024, pp. 13059–13067.
[48] T. Kapoor, A. Chandra, D. Tartakovsky, H. Wang, A. Núñez, R. Dollevoet, Neural oscillators for generalizing parametric PDEs, in: The Symbiosis of Deep Learning and Differential Equations III, 2023.
[49] S. Goswami, C. Anitescu, S. Chakraborty, T. Rabczuk, Transfer learning enhanced physics informed neural network for phase-field modeling of fracture, Theor. Appl. Fract. Mech. 106 (2020) 102447.
[50] T. Kapoor, H. Wang, A. Núñez, R. Dollevoet, Transfer learning for improved generalizability in causal physics-informed neural networks for beam simulations, Eng. Appl. Artif. Intell. 133 (2024) 108085.
[51] M.R.Z. Kouhpanji, B.J. Stadler, First-order reversal curve (FORC) measurements for decoding mixtures of magnetic nanowires, Magn. Meas. Tech. Mater. Charact. (2021) 651–663.
[52] D. Gilbert, FORC diagrams in magnetic thin films, Magn. Meas. Tech. Mater. Charact. (2021) 629–650.
[53] A. Stancu, Characterization of magnetic nanostructures with the first-order reversal curves (FORC) diagram technique, Magn. Meas. Tech. Mater. Charact. (2021) 605–628.
[54] E.S. Loscar, E.V. Albano, Hysteretic effects in the first-order irreversible phase transition of the ZGB model, Comput. Phys. Commun. 180 (4) (2009) 488–492.
[55] T. Kapoor, H. Wang, A. Núñez, R. Dollevoet, Predicting traction return current in electric railway systems through physics-informed neural networks, in: 2022 IEEE Symposium Series on Computational Intelligence (SSCI), 2022, pp. 1460–1468.
[56] B. Daniels, T. Overboom, M. Curti, E. Lomonova, Everett map construction for modeling static hysteresis: Delaunay-based interpolant versus B-spline surface, IEEE Trans. Magn. 59 (5) (2023) 1–4.
[57] B. Daniels, R. Zeinali, T. Overboom, M. Curti, E. Lomonova, Constrained B-spline based Everett map construction for modeling static hysteresis behavior, arXiv preprint, arXiv:2410.02797, 2024.
[58] F. Fiorillo, Measurements of magnetic materials, Metrologia 47 (2) (2010) S114.
[59] T. Kapoor, H. Wang, A. Núñez, R. Dollevoet, Physics-informed neural networks for solving forward and inverse problems in complex beam systems, IEEE Trans. Neural Netw. Learn. Syst. 35 (5) (2024) 5981–5995.
[60] A. He, A. Wang, S. Yue, C. Zhao, C. Chang, H. Men, X. Wang, R.-W. Li, Dynamic magnetic characteristics of $Fe_{78}Si_{13}B_9$ amorphous alloy subjected to operating temperature, J. Magn. Magn. Mater. 408 (2016) 159–163.
[61] I. Reinholds, V. Kalkis, R.D. Maksimovs, The effect of ionizing radiation and magnetic field on deformation properties of high density polyethylene/acrylonitrile-butadiene composites, J. Chem. Chem. Eng. 6 (3) (2012).
[62] Y. Mori, Y. Sakaguchi, H. Hayashi, Magnetic field effects on chemical reactions of biradical radical ion pairs in homogeneous fluid solvents, J. Phys. Chem. A 104 (21) (2000) 4896–4905.
[63] T. Kapoor, H. Wang, A. Núñez, R. Dollevoet, Physics-informed machine learning for moving load problems, J. Phys. Conf. Ser. 2647 (15) (2024) 152003.
[64] T. De Ryck, S. Mishra, R. Molinaro, wPINNs: weak physics informed neural networks for approximating entropy solutions of hyperbolic conservation laws, SIAM J. Numer. Anal. 62 (2) (2024) 811–841.
[65] R. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc., Ser. B Stat. Methodol. 58 (1) (1996) 267–288.
[66] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.