



Deconvolution of electrochemical impedance spectroscopy data using the deep-neural-network-enhanced distribution of relaxation times

Emanuele Quattrochi ^a, Baptiste Py ^a, Adeleke Maradesa ^a, Quentin Meyer ^b, Chuan Zhao ^b, Francesco Ciucci ^{a,c,d,*}

^a Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China

^b School of Chemistry, The University of New South Wales, Sydney, New South Wales 2052, Australia

^c HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen 51048, China

^d Energy Institute, The Hong Kong University of Science and Technology, Hong Kong SAR, China

ARTICLE INFO

Keywords:

Electrochemical impedance spectroscopy
Distribution of relaxation times
Deep neural network
Error analysis
Lithium-metal batteries
Fuel cells

ABSTRACT

Electrochemical impedance spectroscopy (EIS) is widely used to characterize electrochemical systems. The distribution of relaxation times (DRT) has emerged as a powerful, non-parametric alternative to circumvent the inherent challenges of EIS analysis through equivalent circuits or physical models. Recently, deep neural networks have been developed to estimate the DRT. However, this line of research is still in its infancy, and several issues remain unresolved, including the long training time and unknown accuracy of this method. Furthermore, deep neural networks have not been used for deconvolving DRTs with negative peaks. This work addresses these challenges. A pretraining step is included to decrease the computation time; error analysis allows error estimation and the development of error reduction strategies. Furthermore, the training loss function is modified to handle DRTs with negative peaks. For most cases tested, this new framework outperforms ridge regression. Moreover, these advances are validated with an array of synthetic and real EIS spectra from various applications, including lithium-metal batteries, solid oxide fuel cells, and proton exchange membrane fuel cells. Overall, this research opens new avenues for the development and application of the deep-neural-network-based analysis of EIS data.

1. Introduction

Electrochemical impedance spectroscopy (EIS) is an experimental technique widely used for the analysis of electrochemical (e.g., batteries [1], fuel cells [2], and supercapacitors [3]), medical [4], and biological [5] systems. The major appeals of this technique are that it can be carried out over a wide range of frequencies (from mHz to MHz) [6] and that modern equipment allows quick and efficient EIS measurements with limited user expertise. To interpret EIS data, equivalent circuits and physical models are often utilized [7,8]. Despite their simplicity, equivalent circuit models (ECMs) are not unique. Physical models can provide more insights, but are problem-specific and significantly more difficult to implement [9,10]. Considering these challenges, the distribution of relaxation times (DRT) has recently emerged as an alternative approach to analyzing EIS data [11,12]. The DRT model assumes that

the electrochemical processes underpinning EIS spectra are relaxations [13]. This means that the DRT impedance model, $Z_{\text{DRT}}(f, \gamma(\log \tau), L_0, R_\infty)$, at a given frequency, f , is obtained using the following expression [14, 15]:

$$Z_{\text{DRT}}(f, \gamma(\log \tau), L_0, R_\infty) = i2\pi f L_0 + R_\infty + \int_{-\infty}^{+\infty} \frac{\gamma(\log \tau)}{1 + i2\pi f \tau} d\log \tau \quad (1)$$

where L_0 , R_∞ , τ , and $\gamma(\log \tau)$ are the inductance, ohmic resistance, timescale, and DRT, respectively. Usually, the constraint $\gamma(\log \tau) \geq 0$ is also assumed to avoid negative relaxations [16,17].

Despite the DRT's considerable promise, deconvolving $\gamma(\log \tau)$ from experimental data is still challenging. In fact, the inverse problem that needs to be solved to obtain $\gamma(\log \tau)$ is ill-posed with a strong dependence on experimental errors and cut-off frequencies [18–20]. To estimate

Abbreviations: DNN, Deep neural network; DRT, Distribution of relaxation times; ECM, Equivalent circuit model; EIS, Electrochemical impedance spectroscopy; PPD, Points per decade; PWC, Piecewise constant; RR, Ridge regression.

* Corresponding author at: Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China.

E-mail address: francesco.ciucci@ust.hk (F. Ciucci).

<https://doi.org/10.1016/j.electacta.2022.141499>

Received 20 July 2022; Received in revised form 24 October 2022; Accepted 5 November 2022

Available online 7 November 2022

0013-4686/© 2022 Published by Elsevier Ltd.

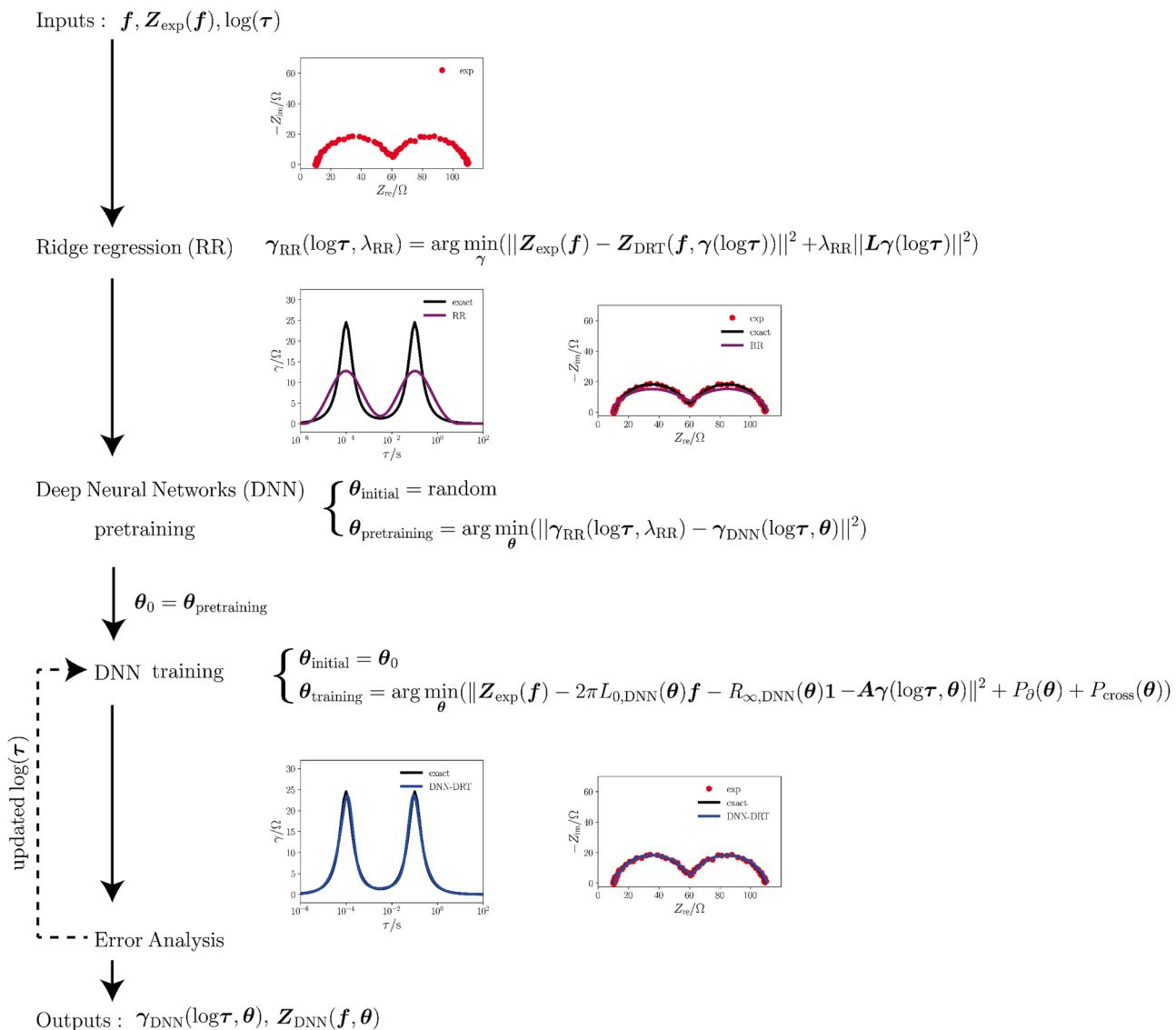


Fig. 1. First, RR is conducted to estimate a coarse DRT, $\gamma_{\text{RR}}(\log\tau, \lambda_{\text{RR}})$, with $\log\tau$ being the vector of log-timescales used for DRT discretization. $\gamma_{\text{RR}}(\log\tau, \lambda_{\text{RR}})$ is then used to pretrain the vector of hyperparameters $\theta_{\text{initial}} = \theta_{\text{pretraining}}$. Full training further optimizes $\gamma_{\text{DNN}}(\log\tau, \theta_{\text{initial}})$. Lastly, error analysis is conducted and, if needed, the vector of log-timescales, $\log\tau$, is updated to improve DNN accuracy.

$\gamma(\log\tau)$, methods leveraging Fourier transform [21,22], genetic algorithms [23,24], Monte Carlo samplers [25,26], and ridge regression (RR) [16,27,28] have been developed. In particular, RR seeks to minimize the following loss function [16,28]:

$$\mathcal{L}(\gamma(\log\tau)) = \|Z_{\text{exp}}(f) - Z_{\text{DRT}}(f, \gamma(\log\tau), L_0, R_\infty)\|^2 + P(\gamma(\log\tau)) \quad (2)$$

where $\gamma(\log\tau)$, $\log\tau$, $Z_{\text{exp}}(f)$, f , and $Z_{\text{DRT}}(f, \gamma(\log\tau), L_0, R_\infty)$ are the vectors of discretized DRTs, log-timescales used to approximate (1), experimental impedances, experimental frequencies, and DRT impedances, respectively, $\|\cdot\|$ is the Euclidean 2-norm, and $P(\gamma(\log\tau))$ is a penalty function. The minimization of (2) is often accompanied by the constraint $\gamma \geq 0$.

It was recently shown that deep neural networks (DNNs) can be utilized to deconvolve the DRT [29,30]. In particular, the first DNN-based method ever developed, called deep-prior DRT, used a DNN with a single random scalar as input [29] and outputted a vector of DRT values at discrete, logarithmically spaced timescales together with the circuit parameters L_0 and R_∞ . The non-negativity constraint (*i.e.*, $\gamma(\log\tau) \geq 0$) illustrated above was imposed by choosing non-negative activation functions in the DNN architecture. Unlike conventional RR,

regularization was not necessary to obtain $\gamma(\log\tau)$ [28,31]. A particularly interesting aspect of the deep-prior DRT approach is that the corresponding inverse problem is overparameterized. In other words, the size of the vector of DNN parameters, including weights and biases, is significantly larger ($\sim 10^4$ parameters) than the number of probed frequencies ($\sim 10^2$ frequencies). As illustrated by the “double-descent” bias-variance curve [32], overparameterization ensures that test errors are small, thereby improving the DNN performance. Quattrochi et al. further extended the deep-prior DRT approach with the deep-DRT model by using a DNN that takes as inputs the scalar log-timescale $\log\tau$ and a state vector, ψ , mapped to experimental conditions (*e.g.*, temperature, pressure, etc.) [30]. In addition to inverting the DRT and regressing the experimental impedance, the trained deep-DRT model was used to predict the DRT and corresponding impedance at experimental states untested experimentally.

While the literature highlights the considerable potential of DNN-based methods for DRT deconvolution, several shortcomings need to be resolved for enhanced EIS analysis:

- (i) DNN training often requires tens of thousands of iterations, which is time-consuming [33,34].
- (ii) Even though the DNN is a universal approximator (see Section S1 of the supplementary information (SI)), DRT deconvolution has been performed so far with unknown accuracy because the integral in (1) was approximated given an arbitrary number of collocation points [28,29].
- (iii) The recovery of DRTs with negative peaks has yet to be studied as the output of DNN methods has been constrained to be non-negative by using a non-negative activation function at the DNN's last layer [29,30].

This article aims to address these challenges. To reduce the computational time, the DNN-DRT model developed herein includes a pre-training step. As shown in Fig. 1, RR is used to obtain a coarse DRT estimate and pretrain the DNN [35]. Then, the accuracy of the pre-trained DNN is improved by minimizing the loss function (2) (generally without the penalty $P(\gamma(\log\tau))$). Second, postprocessing is used to analyze errors, and, if needed, the obtained model error is minimized by first refining the grid over which the integral in (1) is computed and then, if required, by rerunning DNN training. By removing the non-negativity constraint and introducing suitable penalty terms, the DNN-DRT model can accurately recover DRTs with negative peaks. The consistency of the DNN-DRT method was evaluated using synthetic experiments, including EIS data with overlapping features, discontinuities, and negative peaks. Furthermore, the DNN-DRT method was validated against real EIS data from a lithium-metal battery [36], a solid oxide fuel cell [37], and a proton exchange membrane fuel cell [38].

By overcoming these three fundamental limitations (*i.e.*, long training time, DRT discretization with a fixed number of collocation points, and difficulty in deconvolving DRTs with negative peaks), this work paves the way for an enhanced, DNN-based analysis of EIS spectra.

2. Methods

2.1. The DNN-DRT model

The DRT encoded into a DNN is denoted as $\gamma_{\text{DNN}}(\log\tau, \theta)$ with θ being the vector of DNN parameters (*i.e.*, weights and biases). To compute the integral in (1) numerically, the bounds at $\pm\infty$ are replaced by two log-timescales, $\log\tau_{\min}$ and $\log\tau_{\max}$. Using N collocation points, the truncated integral is evaluated numerically over the range of log-timescales, $\log\tau = (\log\tau_1, \log\tau_2, \dots, \log\tau_N)^{\top}$, where $\tau_1 = \tau_{\min}$, $\tau_N = \tau_{\max}$, and $\tau_1 < \tau_2 < \dots < \tau_N$. Throughout the article, these log-timescales are assumed log equispaced, *i.e.*, $\frac{\tau_2}{\tau_1} = \frac{\tau_3}{\tau_2} = \dots = \frac{\tau_N}{\tau_{N-1}}$.

Using the midpoint rule to compute (1), the impedance $Z_{\text{DNN}}(f, \theta)$ at a frequency f is given by [28]

$$\begin{aligned} Z_{\text{DNN}}(f, \theta) &= i2\pi f L_0 + R_{\infty} + \frac{1}{2(1+i2\pi f\tau_1)}\gamma_{\text{DNN}}(\log\tau_1, \theta)\log\frac{\tau_2}{\tau_1} \\ &+ \sum_{n=2}^{N-1}\frac{1}{2(1+i2\pi f\tau_n)}\gamma_{\text{DNN}}(\log\tau_n, \theta)\log\frac{\tau_{n+1}}{\tau_{n-1}} \\ &+ \frac{1}{2(1+i2\pi f\tau_N)}\gamma_{\text{DNN}}(\log\tau_N, \theta)\log\frac{\tau_N}{\tau_{N-1}} \end{aligned} \quad (3)$$

where L_0 and R_{∞} are estimated concomitantly with DNN training.

Given the vector of experimental impedances, $Z_{\text{exp}}(f)$, probed at the M frequencies $f = (f_1, f_2, \dots, f_M)^{\top}$, a latent DRT vector $\gamma_{\text{DNN}}(\log\tau, \theta) = (\gamma_{\text{DNN}}(\log\tau_1, \theta), \gamma_{\text{DNN}}(\log\tau_2, \theta), \dots, \gamma_{\text{DNN}}(\log\tau_N, \theta))^{\top}$ at the log-timescales $\log\tau$ is used to discretize (1) [28], leading to

$$Z_{\text{DNN}}(f, \theta) = i2\pi f L_0 + R_{\infty} \mathbf{1} + (\mathbf{A}_{\text{re}} + i\mathbf{A}_{\text{im}})\gamma_{\text{DNN}}(\log\tau, \theta) \quad (4)$$

where $Z_{\text{DNN}}(f, \theta) = (Z_{\text{DNN}}(f_1, \theta), Z_{\text{DNN}}(f_2, \theta), \dots, Z_{\text{DNN}}(f_M, \theta))^{\top}$ is the vector of DNN impedances at frequencies f_1, f_2, \dots, f_M , $\mathbf{1} = (1, 1, \dots, 1)^{\top} \in \mathbb{R}^M$,

and the matrices \mathbf{A}_{re} and $\mathbf{A}_{\text{im}} \in \mathbb{R}^{M \times N}$ are given in the literature [28].

2.2. Errors

It is assumed that the experimental impedance $Z_{\text{exp}}(f) = (Z_{\text{exp}}(f_1), Z_{\text{exp}}(f_2), \dots, Z_{\text{exp}}(f_M))^{\top}$ is given by

$$Z_{\text{exp}}(f) = Z_{\text{DNN}}(f, \theta) + \epsilon_{\text{epist}} + \epsilon_{\text{trunc}} + \epsilon_{\text{integ}} + \epsilon_{\text{meas}} \quad (5)$$

where ϵ_{epist} , ϵ_{trunc} , ϵ_{integ} , and ϵ_{meas} are the vectors of epistemic, truncation, integration, and measurement errors, respectively [20,39]. The epistemic error, ϵ_{epist} , which is due to model misrepresentation, is not considered as this is beyond the scope of the present work [40]. The numerical errors are due to truncation, ϵ_{trunc} , and integration, ϵ_{integ} (see Section 2.4). Finally, EIS data is inherently laden with random experimental errors, ϵ_{meas} . Hereon, ϵ_{meas} is assumed to be normally distributed, *i.e.*, $\epsilon_{\text{meas}} \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I})$ (σ_n is a scalar and \mathbf{I} the $M \times M$ identity matrix) [41].

2.3. DRT model

2.3.1. Ridge regression

The DRT vector from RR, $\gamma_{\text{RR}}(\log\tau, \lambda_{\text{RR}}) = (\gamma_{\text{RR}}(\log\tau_1, \lambda_{\text{RR}}), \gamma_{\text{RR}}(\log\tau_2, \lambda_{\text{RR}}), \dots, \gamma_{\text{RR}}(\log\tau_N, \lambda_{\text{RR}}))^{\top}$, is obtained by solving the following problem:

$$\begin{aligned} \gamma_{\text{RR}}(\log\tau, \lambda_{\text{RR}}) &= \underset{\gamma \geq 0}{\operatorname{argmin}} \left(\right. \\ &\quad \left. \|Z_{\text{exp}}(f) - (\mathbf{A}_{\text{re}} + i\mathbf{A}_{\text{im}})\gamma(\log\tau)\|^2 + \lambda_{\text{RR}} \|\mathbf{L}\gamma(\log\tau)\|^2 \right) \end{aligned} \quad (6)$$

where \mathbf{L} is a differentiation matrix and $\lambda_{\text{RR}} \geq 0$ is a scalar [28].

2.3.2. DNN deconvolution

To obtain $\gamma_{\text{DNN}}(\log\tau, \theta)$, pretraining is first carried out (see Fig. 1). That is, the weights and biases, $\theta_{\text{pretraining}}$, are obtained by solving

$$\theta_{\text{pretraining}} = \underset{\theta}{\operatorname{argmin}} \|\gamma_{\text{RR}}(\log\tau, \lambda_{\text{RR}}) - \gamma_{\text{DNN}}(\log\tau, \theta)\|^2 \quad (7)$$

where $\gamma_{\text{DNN}}(\log\tau, \theta)$ must be discretized at the same collocation points $\log\tau$ as $\gamma_{\text{RR}}(\log\tau, \lambda_{\text{RR}})$.

Subsequently, the model $\gamma_{\text{DNN}}(\log\tau, \theta)$ is optimized by minimizing

$$\mathcal{L}_{\text{training}}(\theta) = \|Z_{\text{exp}}(f) - i2\pi f L_0 - R_{\infty} \mathbf{1} - (\mathbf{A}_{\text{re}} + i\mathbf{A}_{\text{im}})\gamma_{\text{DNN}}(\log\tau, \theta)\|^2 \quad (8)$$

with respect to θ , where $\theta_{\text{pretraining}}$ is taken as the starting point of the minimization (Fig. 1). It should be noted that the log-timescale vector $\log\tau$ and the matrices \mathbf{A}_{re} and \mathbf{A}_{im} can be updated following RR (see Section 2.5.3 for more details).

2.3.3. Distribution of relaxation times for systems with "inductive" behavior

EIS spectra with "inductive" behavior, *e.g.*, the "hook" [38,42]¹, have DRTs with negative peaks. To account for this, the non-negativity constraint was removed from RR ((6) without the $\gamma \geq 0$ constraint). Furthermore, two penalty terms $P_{\partial}(\theta)$ and $P_{\text{cross}}(\theta)$ were added to the training loss (8), leading to the following modified loss function:

$$\begin{aligned} \mathcal{L}_{\text{training}}(\theta) &= \|Z_{\text{exp}}(f) - i2\pi f L_0 - R_{\infty} \mathbf{1} - (\mathbf{A}_{\text{re}} + i\mathbf{A}_{\text{im}})\gamma_{\text{DNN}}(\log\tau, \theta)\|^2 + P_{\partial}(\theta) \\ &+ P_{\text{cross}}(\theta) \end{aligned} \quad (9)$$

In the latter equation, $P_{\partial}(\theta)$ is defined as

¹ The term "hook" has been associated by Klotz with a negative differential capacitance [42].

Table 1

Truncation, integration, RR, and DNN errors, and corresponding frequency averages.

Error type	Frequency-dependent	Frequency-averaged
Truncation	$\epsilon_{\text{trunc}}(f) = \frac{ \epsilon_{\text{trunc},\text{re}}(f) + i\epsilon_{\text{trunc},\text{im}}(f) }{ Z_{\text{exp}}(f) }$	$\bar{\epsilon}_{\text{trunc}} = \frac{1}{M} \sum_{m=1}^M \epsilon_{\text{trunc}}(f_m)$
Integration	$\epsilon_{\text{integ}}(f) = \frac{ \epsilon_{\text{integ},\text{re}}(f) + i\epsilon_{\text{integ},\text{im}}(f) }{ Z_{\text{exp}}(f) }$	$\bar{\epsilon}_{\text{integ}} = \frac{1}{M} \sum_{m=1}^M \epsilon_{\text{integ}}(f_m)$
RR ¹	$\epsilon_{Z,\text{RR}}(f) = \frac{ Z_{\text{exp}}(f) - Z_{\text{RR}}(f, \lambda_{\text{RR}}) }{ Z_{\text{exp}}(f) }$	$\bar{\epsilon}_{Z,\text{RR}} = \frac{1}{M} \sum_{m=1}^M \epsilon_{Z,\text{RR}}(f_m)$
DNN ²	$\epsilon_{Z,\text{DNN}}(f) = \frac{ Z_{\text{exp}}(f) - Z_{\text{DNN}}(f, \theta) }{ Z_{\text{exp}}(f) }$	$\bar{\epsilon}_{Z,\text{DNN}} = \frac{1}{M} \sum_{m=1}^M \epsilon_{Z,\text{DNN}}(f_m)$

¹ We note that $Z_{\text{RR}}(f)$ is the impedance recovered after RR as the matrix multiplication $Z_{\text{RR}}(f, \lambda_{\text{RR}}) = i2\pi L_0 f + R_\infty \mathbf{1} + (\mathbf{A}_{\text{re}} + i\mathbf{A}_{\text{im}})\gamma_{\text{RR}}(\log\tau, \lambda_{\text{RR}})$ (see (6)).

² For the sake of simplicity, the dependency on θ in $\epsilon_{Z,\text{DNN}}(f)$ is omitted.

$$P_\theta(\theta) = \int_{\log\tau_{\min}}^{\log\tau_{\max}} \lambda(\log\tau, \theta) \left(\frac{\partial^q}{\partial \log\tau^q} \gamma_{\text{DNN}}(\log\tau, \theta) \right)^2 d\log\tau \quad (10)$$

where $q \in \mathbb{Z}^+$ and $\lambda(\log\tau, \theta)$ is a hyper-Bayesian coefficient, see Section S2 for details [13].

The function $P_{\text{cross}}(\theta)$ penalizes the presence of sign changes on $\gamma_{\text{DNN}}(\log\tau, \theta)$ and is given by

$$\begin{aligned} P_{\text{cross}}(\theta) = & \alpha_1 \sum_{n=1}^{N-1} \min(\max(-\gamma_{\text{DNN}}(\log\tau_{n+1}, \theta) \gamma_{\text{DNN}}(\log\tau_n, \theta), 0), \beta_1) \\ & + \alpha_2 \sum_{n=1}^{N-2} \min(\max(-\gamma_{\text{DNN}}(\log\tau_{n+2}, \theta) \gamma_{\text{DNN}}(\log\tau_n, \theta), 0), \beta_2) \end{aligned} \quad (11)$$

where $\log\tau_n$ is the n^{th} log-timescale for $n = 1, 2, \dots, N$, and the four scalars $\alpha_1, \alpha_2, \beta_1$, and β_2 are set to be $\alpha_1 = \frac{\log(\frac{\tau_N}{\tau_1})}{2(N-1)}$, $\alpha_2 = \frac{\alpha_1}{2}$, $\beta_1 = \frac{1}{\alpha_1}$, and $\beta_2 = 2\beta_1$ (see Section S3 for more details).

2.4. Quality assessment through error indicators

To evaluate the quality of the DRT deconvolution and impedance recovery, various error metrics were used [30]. The normalized DRT error for RR, $\epsilon_{\gamma,\text{RR}}$, and for DNN training, $\epsilon_{\gamma,\text{DNN}}$, were defined as

$$\epsilon_{\gamma,\text{RR}} = \frac{\|\gamma_{\text{exact}} - \gamma_{\text{RR}}\|}{\|\gamma_{\text{exact}}\|} \quad (12a)$$

Table 2

Analytical DRTs and impedances used in the synthetic experiments where R_{ct} , $R_{\text{ct},1}$, $R_{\text{ct},2}$ are resistances, ϕ , ϕ_1 , and ϕ_2 are dispersion factors, τ_0 , τ_1 , and τ_2 are timescales, and H is the Heaviside function (the Gerischer model is equivalent to the fractal model with $\phi = 5.00 \times 10^{-1}$).

Model	$\gamma_{\text{exact}}(\log\tau)$	$\gamma_{\text{exact}}(f)$	Ref.
Single ZARC	$\frac{R_{\text{ct}}}{2\pi} \frac{\sin((1-\phi)\pi)}{\cosh\left(\phi\log\left(\frac{\tau}{\tau_0}\right)\right) - \cos((1-\phi)\pi)}$	$R_\infty + \frac{R_{\text{ct}}}{1 + (i2\pi f\tau_0)^\phi}$	[10]
2 × ZARC and "hook"	$\frac{R_{\text{ct},1}}{2\pi} \frac{\sin((1-\phi_1)\pi)}{\cosh\left(\phi_1\log\left(\frac{\tau}{\tau_1}\right)\right) - \cos((1-\phi_1)\pi)} + \frac{R_{\text{ct},2}}{2\pi} \frac{\sin((1-\phi_2)\pi)}{\cosh\left(\phi_2\log\left(\frac{\tau}{\tau_2}\right)\right) - \cos((1-\phi_2)\pi)}$	$R_\infty + \frac{R_{\text{ct},1}}{1 + (i2\pi f\tau_1)^{\phi_1}} + \frac{R_{\text{ct},2}}{1 + (i2\pi f\tau_2)^{\phi_2}}$	[10]
Piecewise constant (PWC)	$\frac{R_{\text{ct}}(H(\tau - \tau_0) - H(\tau - \tau_1))}{\log\left(\frac{\tau_1}{\tau_0}\right)}$	$R_\infty + \frac{R_{\text{ct}}}{\log\left(\frac{\tau_1}{\tau_0}\right)} \left(\log\left(1 - \frac{i}{2\pi f\tau_0}\right) - \log\left(1 - \frac{i}{2\pi f\tau_1}\right) \right)$	[13]
Fractal	$\begin{cases} \frac{R_{\text{ct}}}{\pi} \sin(\phi\pi) \left(\frac{\tau}{\tau_0 - \tau}\right)^\phi & \text{if } \tau < \tau_0 \\ 0 & \text{otherwise} \end{cases}$	$R_\infty + \frac{R_{\text{ct}}}{(1 + i2\pi f\tau_0)^\phi}$	[52]
Gerischer	$\begin{cases} \frac{R_{\text{ct}}}{\pi} \sqrt{\frac{\tau}{\tau_0 - \tau}} & \text{if } \tau < \tau_0 \\ 0 & \text{otherwise} \end{cases}$	$R_\infty + \frac{R_{\text{ct}}}{\sqrt{1 + i2\pi f\tau_0}}$	[53]

$$\epsilon_{\gamma,\text{DNN}} = \frac{\|\gamma_{\text{exact}} - \gamma_{\text{DNN}}\|}{\|\gamma_{\text{exact}}\|} \quad (12b)$$

where $\gamma_{\text{exact}}(\log\tau) = (\gamma_{\text{exact}}(\log\tau_1), \gamma_{\text{exact}}(\log\tau_2), \dots, \gamma_{\text{exact}}(\log\tau_N))^T$ is the known DRT at $\log\tau$. We used 81 logarithmically equispaced timescales between $\log_{10}\tau_1 = -6$ and $\log_{10}\tau_N = 2$ (see Section 2.5.1 on RR) to compute $\epsilon_{\gamma,\text{RR}}$ in (12a). Regarding $\epsilon_{\gamma,\text{DNN}}$ in (12b), the bounds $\log\tau_1$ and $\log\tau_N$ of $\log\tau$ were unchanged, and we investigated the values $= M$, $10 M$, and $100 M$ during the error analysis ($M = 81$).

The quality of the impedance recovery was assessed with four frequency-dependent indicators, namely, the truncation, $\epsilon_{\text{trunc}}(f)$, integration, $\epsilon_{\text{integ}}(f)$, RR, $\epsilon_{Z,\text{RR}}(f)$, and DNN training, $\epsilon_{Z,\text{DNN}}(f)$, errors, whose definitions are given in Table 1. The frequency-averaged of $\epsilon_{\text{trunc}}(f)$, $\epsilon_{\text{integ}}(f)$, $\epsilon_{Z,\text{RR}}(f)$, and $\epsilon_{Z,\text{DNN}}(f)$ are indicated as $\bar{\epsilon}_{\text{trunc}}$, $\bar{\epsilon}_{\text{integ}}$, $\bar{\epsilon}_{Z,\text{RR}}$, and $\bar{\epsilon}_{Z,\text{DNN}}$, and are also presented in Table 1. The real and imaginary parts of the truncation ($\epsilon_{\text{trunc},\text{re}}(f)$ and $\epsilon_{\text{trunc},\text{im}}(f)$) and integration ($\epsilon_{\text{integ},\text{re}}(f)$ and $\epsilon_{\text{integ},\text{im}}(f)$) errors are also derived in Sections S3 and S4.

2.5. Implementation

2.5.1. Ridge regression

Given M experimental frequencies, we approximated the integral in (1) at N log-timescales with $f_k = \frac{1}{\tau_k}$ for $k = 1, \dots, M$ ($N = M$) [16,17,28]. The chosen penalty coefficient λ_{RR} was between 1.00×10^{-5} and 1.00×10^{-2} with the specific values used given in Table S1. The problem (6) was solved using CVXPY [43].

2.5.2. DNN pretraining

To deconvolve the DRT, the γ vectors used in (7) were first normalized with respect to the polarization resistance, R_p , which was obtained by integrating $\gamma_{\text{RR}}(\log\tau, \lambda_{\text{RR}})$ from $\log\tau_{\min}$ to $\log\tau_{\max}$ using the trapezoidal rule. Then, pretraining was carried out by minimizing (7) using the Adam algorithm (2000 steps) [44].

2.5.3. DNN training

The DNN parameters, θ , were obtained by minimizing the loss function in (8), where the vector $\gamma_{\text{DNN}}(\log\tau, \theta)$ was normalized with respect to R_p , and $Z_{\text{exp}}(f)$ was replaced by $\frac{1}{R_p} (Z_{\text{exp}}(f) - i2\pi L_0 f - R_\infty \mathbf{1})$ (see Fig. 1). The penalty terms, $P_\theta(\theta)$ and $P_{\text{cross}}(\theta)$, were added to the loss function $\mathcal{L}_{\text{training}}(\theta)$ in (9) only for EIS spectra with the "hook" feature (Section 2.3.3). For $P_\theta(\theta)$ in (10) ($q = 1$), the first derivative of $\gamma_{\text{DNN}}(\log\tau, \theta)$ was computed by backward differentiation [45]. The influence of the grid density on the DRT recovery was evaluated by discretizing $\gamma_{\text{DNN}}(\log\tau, \theta)$ on a logarithmically equispaced grid with $N = M$,

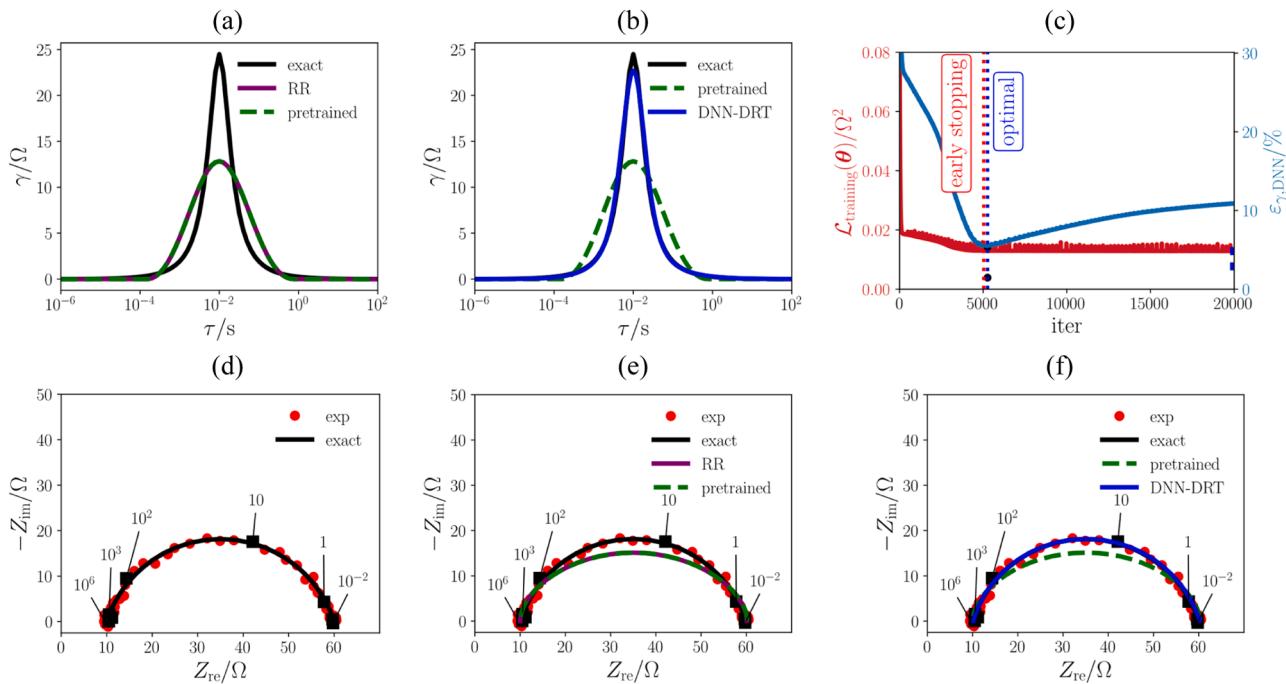


Fig. 2. (a) Exact, RR, and pretrained DRT, and (b) exact, pretrained, and DNN DRT of the single ZARC model ($N = 10 M$); (c) loss function, $\mathcal{L}_{\text{training}}(\theta)$, and DRT error, $\varepsilon_{\gamma,\text{DNN}}$, as functions of the iteration number. Nyquist plots of the (d) experimental and exact impedances, (e) experimental, exact, RR, and pretrained DNN impedances, and (f) experimental, exact, pretrained, and fully trained DNN impedances.

10 M, and 100 M. The error metrics $\varepsilon_{\gamma,\text{RR}}$ and $\varepsilon_{\gamma,\text{DNN}}$ in (12a) and (12b) were computed in the range between $\log\tau_{\min}$ and $\log\tau_{\max}$ ($\tau_1 = \tau_{\min}$, $\tau_N = \tau_{\max}$) utilizing the trapezoidal rule with log equispaced log-timescales, as detailed in Section 2.4.

2.5.4. Deep neural network architecture and optimization

A vanilla feedforward DNN consisting of 12 layers was used, including:

- (i) An input layer of dimension 1 (with the normalized log-timescale $\log\tilde{\tau} = -1 + 2\frac{(\log\tau - \log\tau_{\min})}{(\log\tau_{\max} - \log\tau_{\min})}$ as input).
- (ii) Ten hidden layers of width 32.
- (iii) An output layer of dimension 1 that outputs $\gamma_{\text{DNN}}(\log\tau, \theta)$.

The first layer had a sinusoidal activation function [46]. If the $\gamma(\log\tau) \geq 0$ constraint was enforced, the hidden and output layers had softplus activation functions [47]. When the $\gamma(\log\tau) \geq 0$ constraint was relaxed ("hook" model), non-saturating exponential unit functions were used for the hidden and output layers [48]. The DNN had 10,657 parameters, while L_0 and R_∞ were two additional parameters optimized during training. For DNN pretraining, the weights were initialized using the Xavier method (normal [49] or uniform [50], depending on the layer), and the biases were set to zero or sampled from a normal distribution. More details are provided in Table S2. After DNN pretraining, the obtained pretrained DNN parameters, $\theta_{\text{pretraining}}$, were used as initial values for DNN training (Fig. 1). The learning rates for pretraining and training were set to 1.00×10^{-3} and 1.00×10^{-4} , respectively. To avoid overfitting, an early stopping threshold between two consecutive evaluations of the loss was set at 1.00×10^{-8} . The optimal iteration corresponds to the minimum value of $\varepsilon_{\gamma,\text{DNN}}$. The model was implemented in PyTorch [51].

2.5.5. Generation of the synthetic experiments

Synthetic spectra were generated using the analytical DRTs and impedances reported in Table 2. $M = 81$ log-spaced frequencies between 1.00×10^{-2} and 1.00×10^6 Hz with ten points per decade (ppd) were

used. Unless otherwise specified, the random errors, ϵ_{meas} , were computed according to Section 2.2 with $\sigma_n^{\text{exp}} = 5.00 \times 10^{-1} \Omega$.

2.5.6. Grid density for DRT discretization

A variable number (N) of uniformly log-spaced timescales in the range between $\tau_{\min} = 1.00 \times 10^{-6}$ s and $\tau_{\max} = 1.00 \times 10^2$ s was considered. The minimum, τ_{\min} , and maximum, τ_{\max} , timescales do not vary throughout the article.

3. Results

The DNN-DRT model was first benchmarked against synthetic experiments. First, a single ZARC model was studied. Next, more complex models with overlapping features ($2 \times$ ZARC), discontinuities (fractal, piecewise constant (PWC), Gerischer), and negative peaks ("hook") were considered. Finally, the DNN-DRT model was tested against real EIS data from one battery and two fuel cells.

3.1. Synthetic experiments

3.1.1. Single ZARC model

3.1.1.1 Preliminary analysis. A single ZARC is a circuit comprising a resistor in series with a parallel association of a resistor and a constant phase element (see Table S3 for the parameter values). Fig. 2(a) shows the exact, RR, and pretrained DNN DRTs for $N = 10 M$. RR and pre-training enabled the identification of a single peak. Fig. 2(b) displays the exact, pretrained DNN, and trained DNN DRTs. The DNN DRT recovery was significantly improved compared to RR, as confirmed by the decreased normalized DRT errors ($\varepsilon_{\gamma,\text{RR}} = 4.23 \times 10^{1\%}$ versus $\varepsilon_{\gamma,\text{DNN}} = 5.50 \times 10^{0\%}$, see Table S4). Fig. 2(c) shows $\mathcal{L}_{\text{training}}(\theta)$ (see (8)) and $\varepsilon_{\gamma,\text{DNN}}$ (see (12b)) as functions of the iteration number. With pretraining, $\mathcal{L}_{\text{training}}(\theta)$ and $\varepsilon_{\gamma,\text{DNN}}$ reached their minimum after about 5000 iterations. In contrast, without pretraining they reached their minimum after approximately 8000 iterations, see Fig. S1. Fig. 2(d) shows the experimental and exact impedances, and Fig. 2(e) additionally includes the RR

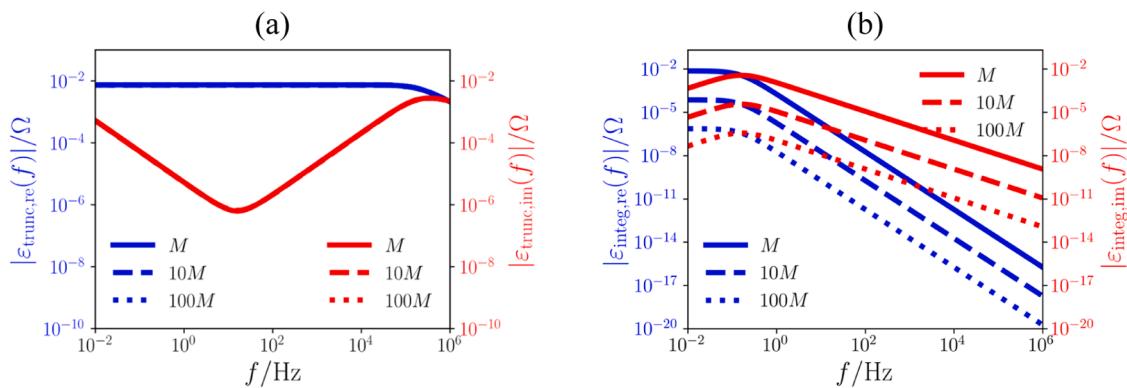


Fig. 3. For the single ZARC model and $N = M$, $10 M$, and $100 M$, absolute values of the real and imaginary components of the (a) truncation and (b) integration errors as functions of the frequency.

and pretrained DNN impedances. The pretraining step led to a DNN impedance closely matching the RR impedance (Fig. 2(e)). The impedance recovery was further improved by the DNN training as shown in Fig. 2(f). This was also confirmed by a reduction in averaged impedance errors ($\bar{\varepsilon}_{Z,\text{RR}} = 5.69 \times 10^0\%$ versus $\bar{\varepsilon}_{Z,\text{DNN}} = 3.38 \times 10^0\%$, Table S5). Repeating this artificial experiment 10 times led to the same results (Tables S6 and S7), confirming that DNN training improved impedance recovery.

Fig. 3(a) shows the absolute value of the two components, $|\varepsilon_{\text{trunc},\text{re}}(f)|$ and $|\varepsilon_{\text{trunc},\text{im}}(f)|$,² of the truncation error as functions of frequency. $|\varepsilon_{\text{trunc},\text{re}}(f)|$ was virtually frequency-independent up to $f \approx 5.00 \times 10^5$ Hz and experienced a slight decrease after that value. Conversely, $|\varepsilon_{\text{trunc},\text{im}}(f)|$ decreased until a minimum was reached at $f \approx 1.00 \times 10^1$ Hz before increasing again up to $f \approx 5.00 \times 10^5$ Hz. It is noticeable that $|\varepsilon_{\text{trunc},\text{im}}(f)| < |\varepsilon_{\text{trunc},\text{re}}(f)|$ for all frequencies. Fig. 3(b) shows the absolute value of the two components $\varepsilon_{\text{integ},\text{re}}(f)$ and $\varepsilon_{\text{integ},\text{im}}(f)$ of the integration error as functions of frequency. While $|\varepsilon_{\text{integ},\text{re}}(f)|$ decreased monotonically from $f \approx 1.00 \times 10^{-1}$ Hz, $|\varepsilon_{\text{integ},\text{im}}(f)|$ reached a maximum at around $f \approx 1.00 \times 10^{-1}$ Hz before later decreasing (see Fig. 3(b)). Consistent with intuition, the truncation errors were insensitive to N , while the integration errors decreased as N increased.

3.1.1.2 Influence of the grid density on the truncation and integration errors. We set $M = 81$ and carried out DNN simulations for $N = M$, $5 M$, $10 M$, ..., $500 M$ (RR was performed with $N = M$ as indicated in Section 2.5.1). For all N , the DNN led to a better DRT recovery relative to RR (see Fig. S2 and the boxplots in Fig. S3). The DRT and impedance errors were also consistently lower for the DNN compared to RR ($\varepsilon_{Y,\text{RR}} = 4.23 \times 10^1\%$ versus $\varepsilon_{Y,\text{DNN}} = 5.50 \times 10^0\%$, $\bar{\varepsilon}_{Z,\text{RR}} = 5.69 \times 10^0\%$ versus $\bar{\varepsilon}_{Z,\text{DNN}} = 3.38 \times 10^0\%$, see Table S8). Fig. 3(a) shows that $|\varepsilon_{\text{trunc},\text{re}}| > |\varepsilon_{\text{trunc},\text{im}}|$ for $N = M$, $10 M$, and $100 M$, and both errors are independent of N . Moreover, $|\varepsilon_{\text{integ},\text{re}}|$ and $|\varepsilon_{\text{integ},\text{im}}|$ decreased when N increased (Fig. 3(b)), consistent with intuition. Similar to Section 3.1.1.1, these conclusions were further reproduced with ten additional synthetic experiments (see Tables S8 and S9).

3.1.1.3 Influence of the experimental noise. In order to study the impact of the experimental noise on the DRT and impedance recoveries, we considered $\sigma_n^{\text{exp}} = 1.00 \times 10^{-1} \Omega$, $2.00 \times 10^0 \Omega$, and $5.00 \times 10^0 \Omega$. When σ_n^{exp} was increased from $1.00 \times 10^{-1} \Omega$ to $5.00 \times 10^0 \Omega$, the RR DRT deconvolution was not affected (see Fig. S4(a), (d), (g), and (k), and $\varepsilon_{Y,\text{RR}} = 4.23 \times 10^1\%$ for $\sigma_n^{\text{exp}} = 5.00 \times 10^{-1} \Omega$ and $N = 10 M$, Table S10) as the RR regularization parameter λ_{RR} (Section 2.3.1) was chosen to be large

enough. However, $\varepsilon_{Y,\text{DNN}}$ increased ($\varepsilon_{Y,\text{DNN}} = 4.38 \times 10^0\%$, $5.50 \times 10^0\%$, $1.75 \times 10^1\%$, and $2.61 \times 10^1\%$ for $\sigma_n^{\text{exp}} = 1.00 \times 10^{-1} \Omega$, $2.00 \times 10^0 \Omega$, and $5.00 \times 10^0 \Omega$, respectively). Additionally, $\mathcal{L}_{\text{training}}(\theta)$ and $\varepsilon_{Y,\text{DNN}}$ reached their minima in fewer iterations when σ_n^{exp} increased, but the minimum of $\mathcal{L}_{\text{training}}(\theta)$ increased (Fig. S4). In turn, the impedance error for both RR and DNN deconvolution increased ($\bar{\varepsilon}_{Z,\text{RR}} = 4.01 \times 10^0\%$, $5.69 \times 10^0\%$, $1.42 \times 10^1\%$, and $3.43 \times 10^1\%$, and $\bar{\varepsilon}_{Z,\text{DNN}} = 7.10 \times 10^{-1}\%$, $3.38 \times 10^0\%$, $1.32 \times 10^1\%$, and $3.40 \times 10^1\%$ for $\sigma_n^{\text{exp}} = 1.00 \times 10^{-1} \Omega$, $5.00 \times 10^{-1} \Omega$, $2.00 \times 10^0 \Omega$, and $5.00 \times 10^0 \Omega$, respectively, see Table S11).³ In addition, the values of $\varepsilon_{Y,\text{RR}}$, $\varepsilon_{Y,\text{DNN}}$, $\bar{\varepsilon}_{Z,\text{RR}}$, and $\bar{\varepsilon}_{Z,\text{DNN}}$ did not significantly change for $N = M$, $10 M$, and $100 M$ (see Tables S10 and S11). Lastly, the conclusions drawn for $\varepsilon_{Y,\text{RR}}$, $\varepsilon_{Y,\text{DNN}}$, $\bar{\varepsilon}_{Z,\text{RR}}$, and $\bar{\varepsilon}_{Z,\text{DNN}}$ for $N = M$ and the various σ_n^{exp} were further validated using ten additional synthetic experiments (Tables S12 and S13).

3.1.1.4 Influence of λ_{RR} on the DNN training. RR requires the use of a penalty parameter, λ_{RR} , see (6), which has significant impact on the obtained DRTs and impedances [16,28,31]. To investigate the effect of λ_{RR} on $\gamma_{\text{RR}}(\log \tau, \lambda_{\text{RR}})$, we first performed RR ($N = M$) for $\lambda_{\text{RR}} = 1.00 \times 10^{-6}$, 1.00×10^{-5} , ..., 1.00×10^{-1} , and then carried out DNN training ($N = 10 M$). For small values of λ_{RR} (i.e., $\lambda_{\text{RR}} = 1.00 \times 10^{-6}$, 1.00×10^{-5} , and 1.00×10^{-4}), oscillations were observed at the outer edges of the DRT peak (Fig. S5), while large values of λ_{RR} (i.e., $\lambda_{\text{RR}} = 1.00 \times 10^{-3}$, 1.00×10^{-2} , and 1.00×10^{-1}) widened and reduced the height of the DRT peak (Fig. S6). In other words, small and large values of λ_{RR} led to underfitting and overfitting the EIS, respectively [28,54]. The trained DNN closely matched the exact DRT for all values of λ_{RR} studied (Figs. S5 and S6), implying that λ_{RR} did not influence DNN performance (see the DNN errors $\varepsilon_{Y,\text{DNN}}$ and $\bar{\varepsilon}_{Z,\text{DNN}}$ in Table S14). These conclusions were further validated using ten additional synthetic experiments (see Tables S14 and S15).

3.1.1.5 Comparison with the deep-DRT model. We benchmarked the DNN-DRT model against the previously developed deep-DRT model ($N = M$). The parameters used for the deep-DRT model are given in Table S16 [30]. The values of $\varepsilon_{Y,\text{DNN}}$ and $\bar{\varepsilon}_{Z,\text{DNN}}$ were lower for the DNN-DRT model ($\varepsilon_{Y,\text{DNN}} = 7.65 \times 10^0\%$ and $\bar{\varepsilon}_{Z,\text{DNN}} = 6.67 \times 10^0\%$, Table S17) than for the deep-DRT model ($\varepsilon_{Y,\text{DNN}} = 8.06 \times 10^0\%$ and $\bar{\varepsilon}_{Z,\text{DNN}} = 8.38 \times 10^0\%$). In other words, the proposed new DNN architecture led to better recoveries of DRT and impedance. These findings were further reproduced with ten additional synthetic experiments (Tables S17 and S18).

² As previously defined (Section 2.4), $\varepsilon_{\text{trunc},\text{re}}(f)$ and $\varepsilon_{\text{trunc},\text{im}}(f)$ are the integral truncation errors on the real and imaginary parts of $Z_{\text{DRT}}(f)$, respectively.

³ This phenomenon was also observed in [16], where a better approximation fares better at low σ_n^{exp} .

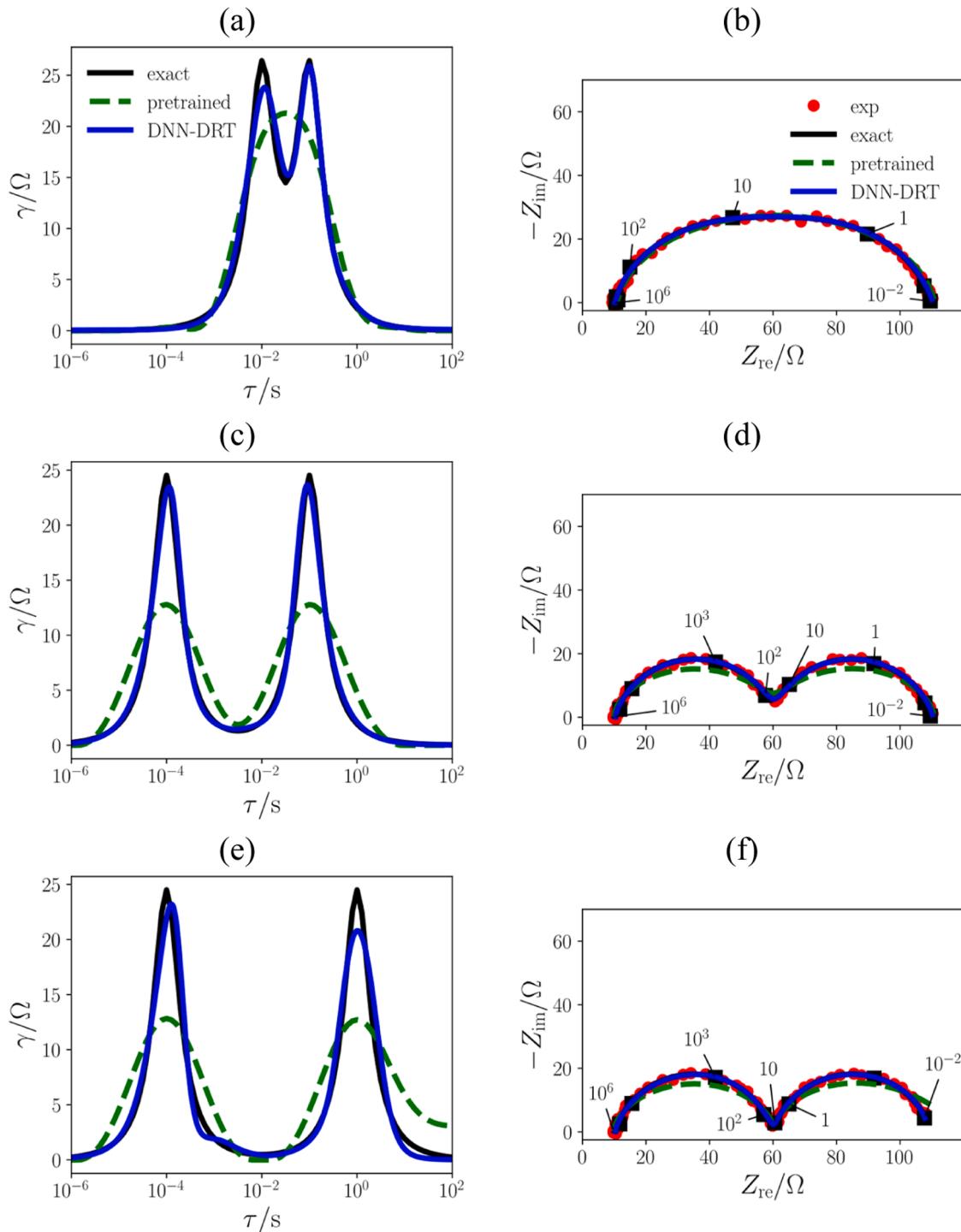


Fig. 4. Exact, pretrained DNN, and trained DNN DRTs ($N = 10 M$) of the (a) overlapping, (c) separated, and (e) distant $2 \times$ ZARC models; corresponding Nyquist plots are reported in panels (b), (d), and (f) with synthetic experimental impedances also shown.

3.1.2. $2 \times$ ZARC model

3.1.2.1 Overlapping, separated, and distant $2 \times$ ZARC models. This section studies the ability of the DNN-DRT model to separate processes characterized by close timescales. The overlapping ($\tau_1 = 1.00 \times 10^{-1} \text{ s}$, $\tau_2 = 1.00 \times 10^{-2} \text{ s}$), separated ($\tau_1 = 1.00 \times 10^{-1} \text{ s}$, $\tau_2 = 1.00 \times 10^{-4} \text{ s}$), and distant ($\tau_1 = 1.00 \times 10^0 \text{ s}$, $\tau_2 = 1.00 \times 10^{-4} \text{ s}$) $2 \times$ ZARC models were first studied (the parameters are given in Table S19). The left panels of Fig. 4 show, for each $2 \times$ ZARC model, the exact, pretrained DNN, and trained DNN DRTs with the corresponding impedances

displayed on the right panels of Fig. 4. As above (Section 3.1.1.1), the trained DNN DRT was much closer to the exact DRT than the pretrained model (Fig. 4). The errors on $\gamma(\log\tau)$ obtained for the overlapping, separated, and distant $2 \times$ ZARC models were greatly reduced after DNN training ($\varepsilon_{\gamma,\text{RR}} = 2.43 \times 10^{1\%}$, $4.20 \times 10^{1\%}$, and $4.45 \times 10^{1\%}$ versus $\varepsilon_{\gamma,\text{DNN}} = 6.57 \times 10^{0\%}$, $6.59 \times 10^{0\%}$, and $1.36 \times 10^{1\%}$, respectively, see Table S4). Moreover, the trained DNN slightly improved the impedance recovery relatively to RR, as shown in Fig. 4; the impedance errors for RR ($\bar{\varepsilon}_{Z,\text{RR}} = 3.28 \times 10^{0\%}$, $5.48 \times 10^{0\%}$, and $5.34 \times 10^{0\%}$, see Table S5) were only slightly above those computed using DNN training ($\bar{\varepsilon}_{Z,\text{DNN}} =$

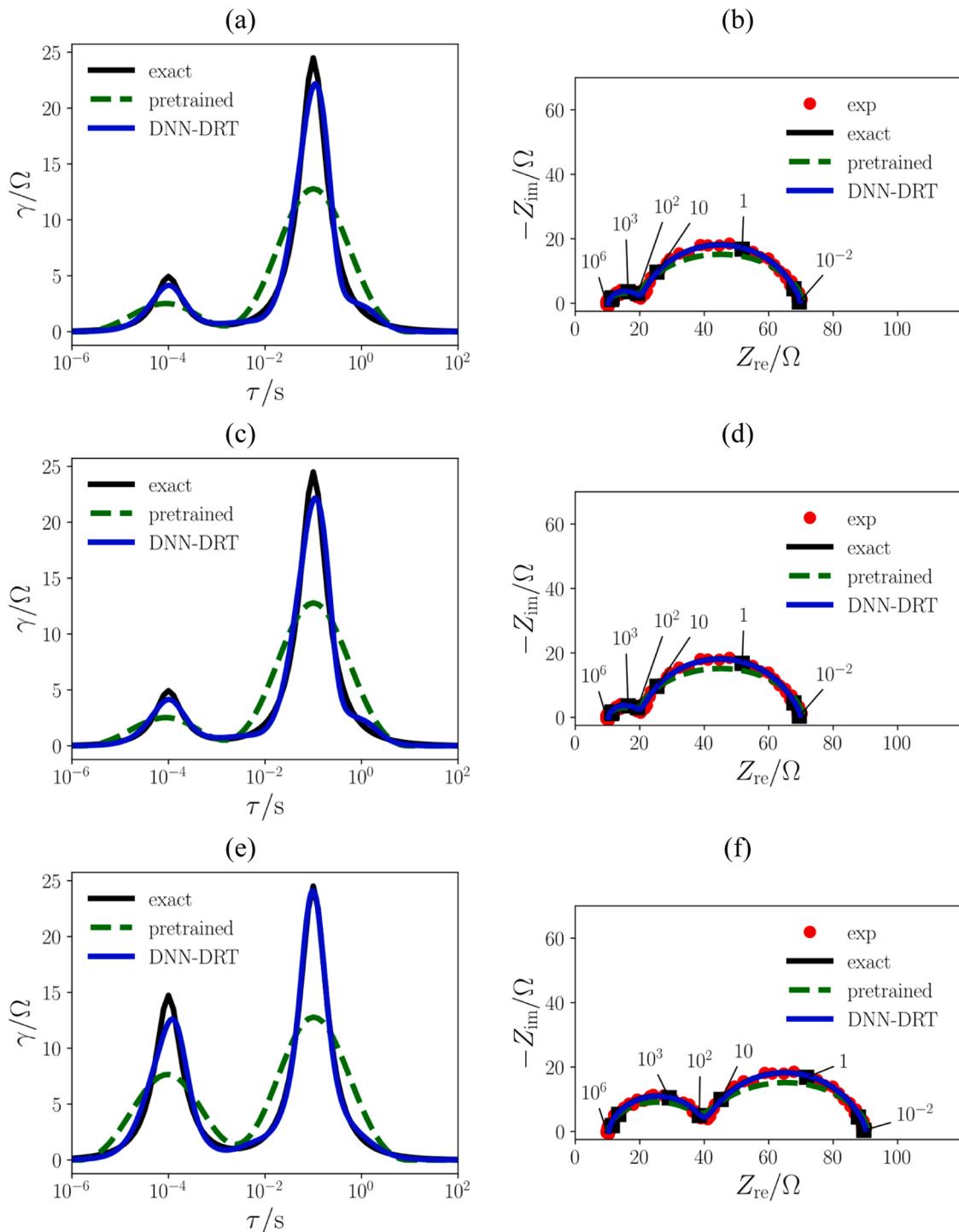


Fig. 5. Exact, pretrained DNN, and trained DNN DRTs ($N = 10 M$) of the separated $2 \times$ ZARC model with $R_{ct,2} =$ (a) $1.00 \times 10^1 \Omega$, (c) $1.00 \times 10^1 \Omega$, and (e) $5.00 \times 10^1 \Omega$; corresponding Nyquist plots are reported in panels (b), (d), and (f) with synthetic experimental impedances also shown.

$3.22 \times 10^0\%$, $2.12 \times 10^0\%$, and $2.08 \times 10^0\%$). The values of both error metrics for $N = M$, $10 M$, and $100 M$ are presented in the boxplots in Fig. S7, and Tables S4 and S5. These conclusions were further reproduced using ten additional synthetic experiments (Tables S6 and S7, and Fig. S7).

As a second step, the DRT deconvolution of the separated $2 \times$ ZARC model with unequal peaks (the parameters can be found in Table S19) was analyzed. The DRTs and Nyquist plots are presented in Fig. 5, and the errors in Table S4. It was found that DNN training improved the recoveries of DRT and impedance compared to RR (the errors $\varepsilon_{\gamma,RR}$,

$\varepsilon_{\gamma,DNN}$, $\bar{\varepsilon}_{Z,RR}$, and $\bar{\varepsilon}_{Z,DNN}$ for $N = M$, $10 M$, and $100 M$ are given in Tables S4 and S5). These conclusions were further validated with ten additional synthetic spectra (Tables S6 and S7).

As a third step, we compared the DNN-DRT model against DRTtools using the separated $2 \times$ ZARC model with peaks of different heights ($R_{ct,1} = 5.00 \times 10^1 \Omega$, $R_{ct,2} = 1.00 \times 10^1 \Omega$ and $3.00 \times 10^1 \Omega$, and $N = M$) [16]. The DRT recovery using the DNN-DRT outperformed DRTtools (see $\varepsilon_{\gamma,DNN}$ in Table S20), but led to a similar impedance recovery, see Fig. S8 and $\bar{\varepsilon}_{Z,DNN}$ in Table S20.

As a fourth step, we used the overlapping and separated $2 \times$ ZARC

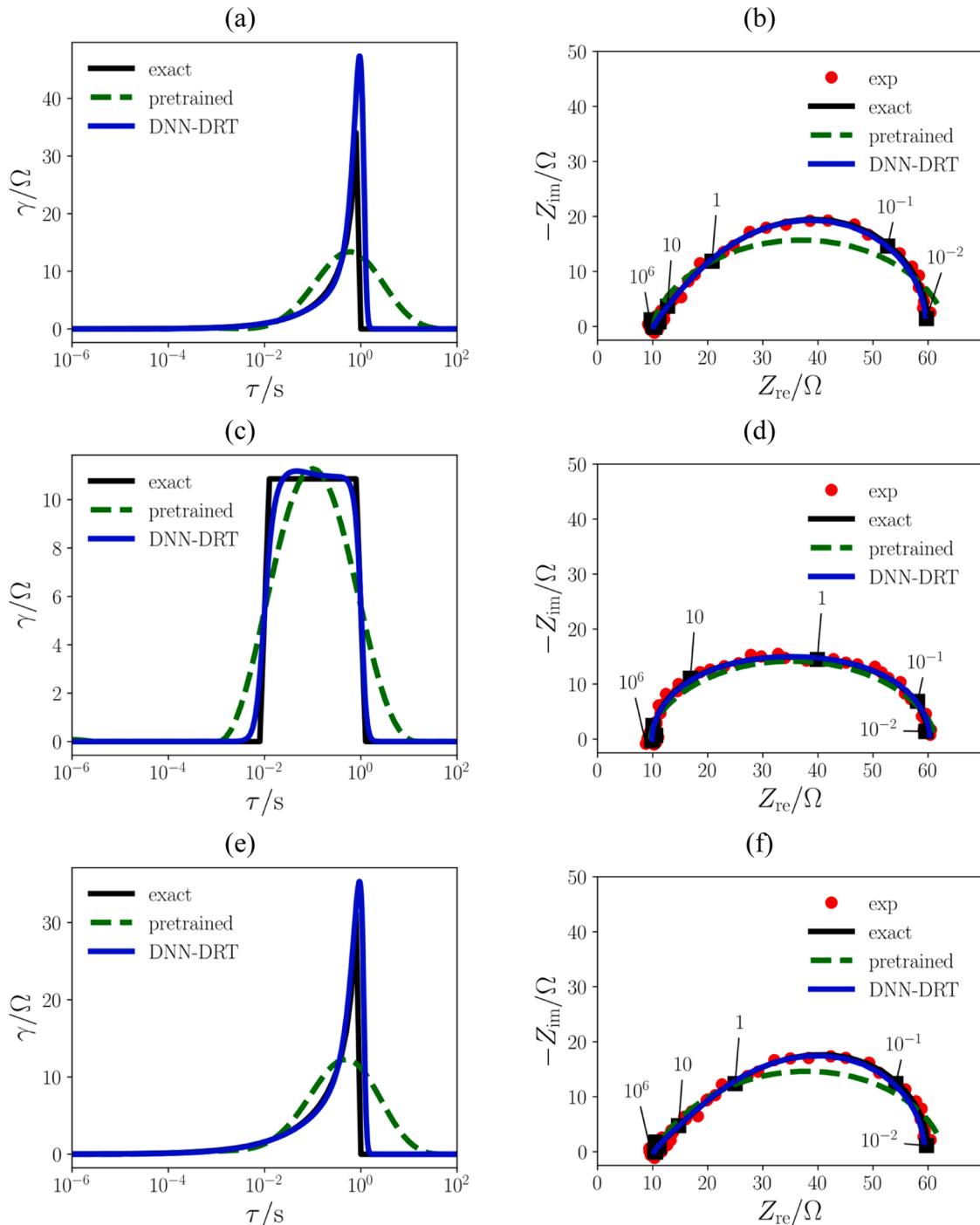


Fig. 6. Exact, pretrained DNN, and trained DNN DRTs ($N = 10 M$) of the (a) fractal, (c) PWC, and (e) Gerischer models; corresponding Nyquist plots are reported in panels (b), (d), and (f) with synthetic experimental impedances also shown.

models with $N = M$ to benchmark the DNN-DRT model against the deep-DRT model (the values of the parameters are given in Table S16 [30]). For the two $2 \times$ ZARC models, the DNN-DRT model led to a better impedance recovery, see Table S17. This finding was further reproduced with ten additional synthetic spectra (Tables S17 and S18).

3.1.2.2 Truncation errors. In this section, we used the separated $2 \times$ ZARC model with peaks of the same height to study the impact of the integration bounds $\tau_1 = \tau_{\min}$ and $\tau_N = \tau_{\max}$ on the truncation error $\bar{\epsilon}_{\text{trunc}}$. For the initial case with $\tau_1 = 1.00 \times 10^{-8} \text{ s}$ and $\tau_N = 1.00 \times 10^4 \text{ s}$, we found that $\bar{\epsilon}_{\text{trunc}} = 9.20 \times 10^{-1}\%$ and $\bar{\epsilon}_{Z,\text{DNN}} = 2.12 \times 10^0\%$ (Table S5)

with $N = 10 M$. By setting $\tau_1 = 1.00 \times 10^{-8} \text{ s}$ and $\tau_N = 1.00 \times 10^4 \text{ s}$, $\bar{\epsilon}_{\text{trunc}}$ was remarkably reduced ($\bar{\epsilon}_{\text{trunc}} = 2.57 \times 10^{-2}\%$, with $N = 10 M$, see Table S21). In other words, extending the range of timescales reduced $\bar{\epsilon}_{\text{trunc}}$ significantly, consistent with intuition.

3.1.3. Discontinuous DRTs

We tested how well the DNN-DRT model ($N = M$, $10 M$, and $100 M$) recovers discontinuous DRTs by studying the fractal, PWC, and Gerischer models (the parameters are given in Tables S22, S23, and S24, respectively). For all three models, the peak was approximately identified by RR (Figs. 6, S9, and S10). Therefore, large values of $\epsilon_{\gamma,\text{RR}}$ were

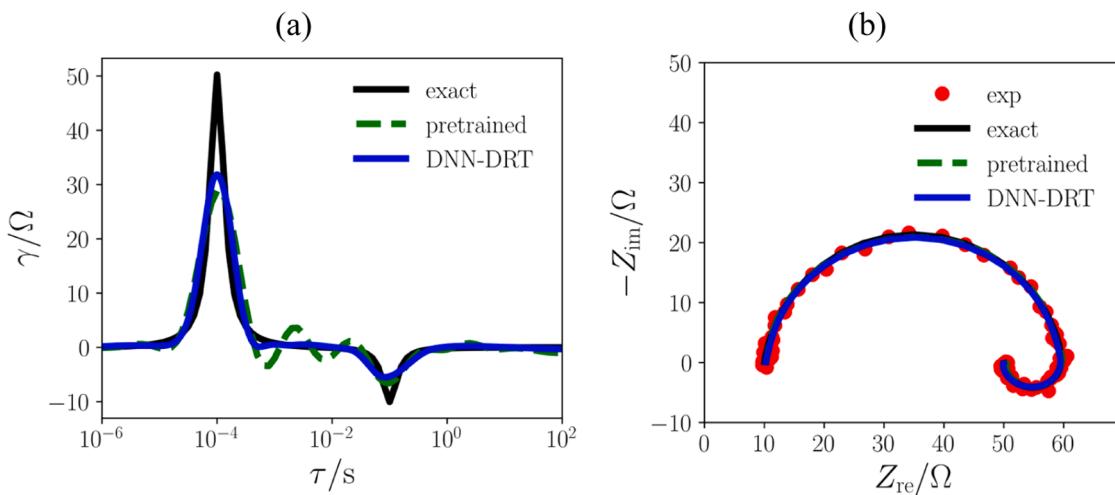


Fig. 7. (a) Exact, pretrained DNN, and trained DNN DRTs ($N = 10 M$) of the “hook” model; (b) corresponding Nyquist plot is reported with the synthetic experimental impedance also shown.

computed ($\epsilon_{\gamma,\text{RR}} = 7.72 \times 10^{1\%}$, $2.95 \times 10^{1\%}$, and $6.80 \times 10^{1\%}$, see Table S4) for the fractal, PWC, and Gerischer models, respectively. Using the DNN increased the accuracy of the DRT recovery ($\epsilon_{\gamma,\text{DNN}} = 6.24 \times 10^{1\%}$, $1.64 \times 10^{1\%}$, and $5.39 \times 10^{1\%}$ for $N = 10 M$). The DNN training also improved the impedance recovery compared to RR (Figs. 6, S9, and S10). This is evidenced by the impedance errors for RR ($\bar{\epsilon}_{Z,\text{RR}} = 6.94 \times 10^{0\%}$, $5.37 \times 10^{0\%}$, and $6.00 \times 10^{0\%}$, see Table S5) and DNN training ($\bar{\epsilon}_{Z,\text{DNN}} = 4.28 \times 10^{0\%}$, $4.15 \times 10^{0\%}$, and $4.16 \times 10^{0\%}$). These conclusions were further reproduced with ten additional synthetic experiments (see the boxplots in Fig. S11, and Tables S6 and S7).

As a second step, we used the fractal, Gerischer, and PWC models to compare the DNN-DRT model ($N = M$) against DRTtools.⁴ For all three impedance models, the DNN-DRT model excelled DRTtools for the DRT recovery, and led to similar impedance recovery except for the fractal model, for which DRTtools outperformed (see Fig. S8 and Table S20).

As a third step, we used the fractal and PWC models to compare the DNN-DRT model ($N = M$) and deep-DRT model (the parameters are reported in Table S16). We found that the DNN-DRT model outperformed the deep-DRT model for the DRT recovery of the fractal and PWC models (Table S17). Moreover, both the DNN-DRT model and deep-DRT model recovered the fractal and PWC impedances with similar accuracy. These findings regarding the performance of the DNN-DRT model and the deep-DRT model were further confirmed with ten additional synthetic spectra (Tables S17 and S18).

3.1.4. DRTs with negative peaks

3.1.4.1 Preliminary analysis. We studied whether the DNN-DRT model can handle DRTs with negative peaks (the parameters are reported in Table S25 [38]). RR led to oscillations at the edges of each peak, with both peak heights being underestimated (Fig. 7(a)). In turn, the RR error $\epsilon_{\gamma,\text{RR}} = 4.13 \times 10^{1\%}$ (Table S4) was large. DNN training ($N = 10 M$) improved the DRT recovery, leading to $\epsilon_{\gamma,\text{DNN}} = 3.18 \times 10^{1\%}$. The specific role of each penalty term $P_\delta(\theta)$ and $P_{\text{cross}}(\theta)$ in $\mathcal{L}_{\text{training}}(\theta)$ in (9) will be further investigated in the next section (Section 3.1.4.2). The impedance recovery was accurate for both RR and DNN-DRT (Fig. 7(b)), as evidenced by the small impedance errors ($\bar{\epsilon}_{Z,\text{RR}} = 2.27 \times 10^{0\%}$ and $\bar{\epsilon}_{Z,\text{DNN}} = 2.36 \times 10^{0\%}$, see Table S5). We stress that the impedance recovery was accurate for all values of N ($N = M$, $10 M$, and $100 M$), see

Fig. S12, and Tables S4 and S5. These different conclusions were further reproduced with ten additional synthetic spectra (see the boxplots in Fig. S11, and Tables S6 and S7).

3.1.4.2 Effect of the penalty functions on the recovery of “inductive” DRTs. For $N = 10 M$, we studied whether the penalties $P_\delta(\theta)$ and $P_{\text{cross}}(\theta)$ suppress oscillations in DNN training to reduce the errors $\epsilon_{\gamma,\text{DNN}}$ and $\bar{\epsilon}_{Z,\text{DNN}}$. The list of investigated cases and the corresponding errors $\epsilon_{\gamma,\text{DNN}}$ and $\bar{\epsilon}_{Z,\text{DNN}}$ are summarized in Table S26. We found that $P_\delta(\theta)$ alone could not suppress the oscillations due to RR (Fig. S13 (a)). Moreover, the errors $\epsilon_{\gamma,\text{DNN}}$ and $\bar{\epsilon}_{Z,\text{DNN}}$ when only $P_\delta(\theta)$ was used ($\epsilon_{\gamma,\text{DNN}} = 3.89 \times 10^{1\%}$ and $\bar{\epsilon}_{Z,\text{DNN}} = 4.15 \times 10^{0\%}$, see Table S26) were close to the case without penalty ($\epsilon_{\gamma,\text{DNN}} = 3.89 \times 10^{1\%}$ and $\bar{\epsilon}_{Z,\text{DNN}} = 3.84 \times 10^{0\%}$, and Fig. S13(b)). In both cases, the “hook” shape at low frequencies was not accurately recovered (panels (c) and (d) of Fig. S13). Additionally, the lowest $\epsilon_{\gamma,\text{DNN}}$ (i.e., the most accurate DRT recovery) was achieved using only the second summation term in $P_{\text{cross}}(\theta)$ ($\epsilon_{\gamma,\text{DNN}} = 3.15 \times 10^{1\%}$ and $\bar{\epsilon}_{Z,\text{DNN}} = 2.40 \times 10^{0\%}$), while the lowest $\bar{\epsilon}_{Z,\text{DNN}}$ (i.e., the most accurate impedance recovery) was obtained by using $P_\delta(\theta)$ and only the first summation term in $P_{\text{cross}}(\theta)$ ($\epsilon_{\gamma,\text{DNN}} = 3.28 \times 10^{1\%}$ and $\bar{\epsilon}_{Z,\text{DNN}} = 2.35 \times 10^{0\%}$).

3.1.5. Summary

Synthetic EIS data, characterized by continuous, overlapping, discontinuous, and negative DRTs, were used to validate the DNN-DRT model. The pretraining step was shown to reduce DNN training steps, resulting in fewer iterations. Additionally, an error analysis was carried out to evaluate the DRT ($\epsilon_{\gamma,\text{DNN}}$) and impedance ($\bar{\epsilon}_{Z,\text{DNN}}$) errors after RR and DNN training. Furthermore, we modified the loss function for the DNN training to handle DRTs with negative peaks. We also used synthetic spectra to show that the DNN-DRT model performs up to three times better than the deep-DRT model and DRTtools. Future work with synthetic EIS data could focus on validating new DNN architectures and on the contribution of the epistemic error to the total DNN error. In the upcoming section (Section 3.2), we further validate the DNN-DRT model with real EIS data.

3.2. Deconvolution of experimental data

We benchmarked the DRT-DNN model using EIS data collected from a lithium-metal battery, a solid oxide fuel cell, and a proton exchange membrane fuel cell.

⁴ For the DRTtools, we used piecewise linear functions for the discretization, included the real and imaginary parts of the impedance, excluded the inductance, took the 2nd derivative for the regularization, and 1.00×10^{-2} for λ_{RR} .

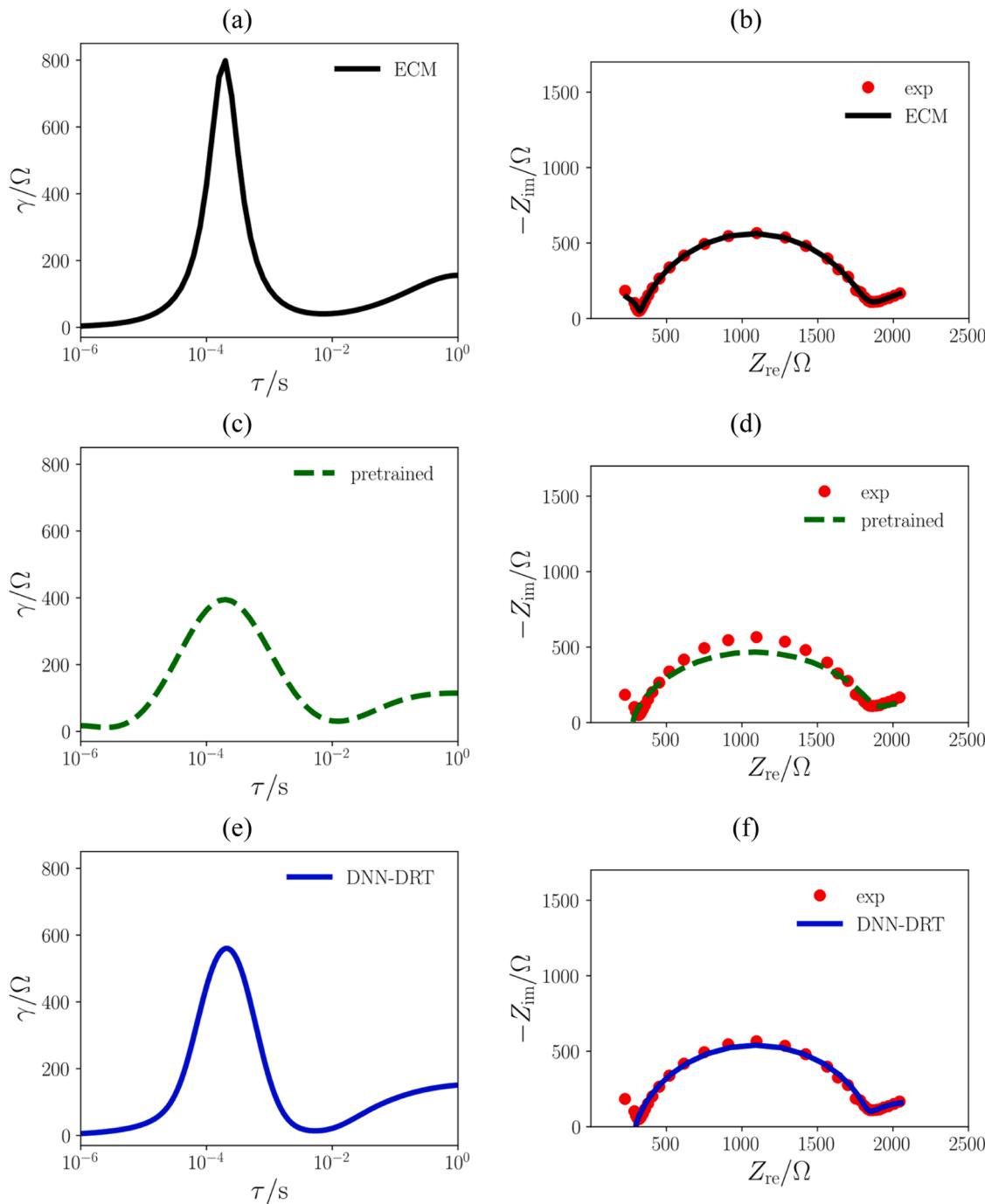


Fig. 8. (a) Fitted-ECM, (c) pretrained DNN, and (e) trained DNN DRTs for a lithium-metal symmetric cell ($N = 100 M$); corresponding Nyquist plots are reported in panels (b), (d), and (f) with the experimental impedances also shown.

3.2.1. Lithium-metal battery

We considered a symmetric cell with lithium-metal electrodes and a composite polymer electrolyte [36]. The impedance was modeled with a $3 \times \text{ZARC}$ (see Table S27 for the parameters). Fig. 8 presents the recovered DRT and impedance, while Fig. S14 shows the plots of the impedance modulus and argument. Regarding the DRT, the peaks at $\tau_1 = 1.03 \times 10^0 \text{ s}$ and $\tau_2 = 1.91 \times 10^{-4} \text{ s}$ were well identified. However, the

peak at $\tau_3 = 1.05 \times 10^{-7} \text{ s}$ ($\phi_3 = 1.00 \times 10^0$) was not recovered. RR yielded a high discrepancy for the DRT with $\epsilon_{\gamma,\text{RR}} = 4.42 \times 10^{1\%}$ ⁵ for $N = M, 10 M$, and $100 M$ (Table S28), which was greatly reduced after DNN training ($\epsilon_{\gamma,\text{DNN}} = 2.62 \times 10^1\%$). The recovered impedance also had higher errors for RR than DNN training ($\bar{\epsilon}_{Z,\text{RR}} = 9.30 \times 10^0\%$ and $\bar{\epsilon}_{Z,\text{DNN}} = 6.06 \times 10^0\%$ for $N = 10M$, see Table S29).

⁵ We stress that the metrics $\epsilon_{\gamma,\text{RR}}$ and $\bar{\epsilon}_{\gamma,\text{DNN}}$ in ((12a) and (12b), respectively, are defined by replacing γ_{exact} with the ECM-based DRT γ_{ECM} . Since ECMS are not unique, no exact value can be given to either $\epsilon_{\gamma,\text{RR}}$ or $\bar{\epsilon}_{\gamma,\text{DNN}}$, unlike for synthetic data, for which γ_{exact} is known exactly.

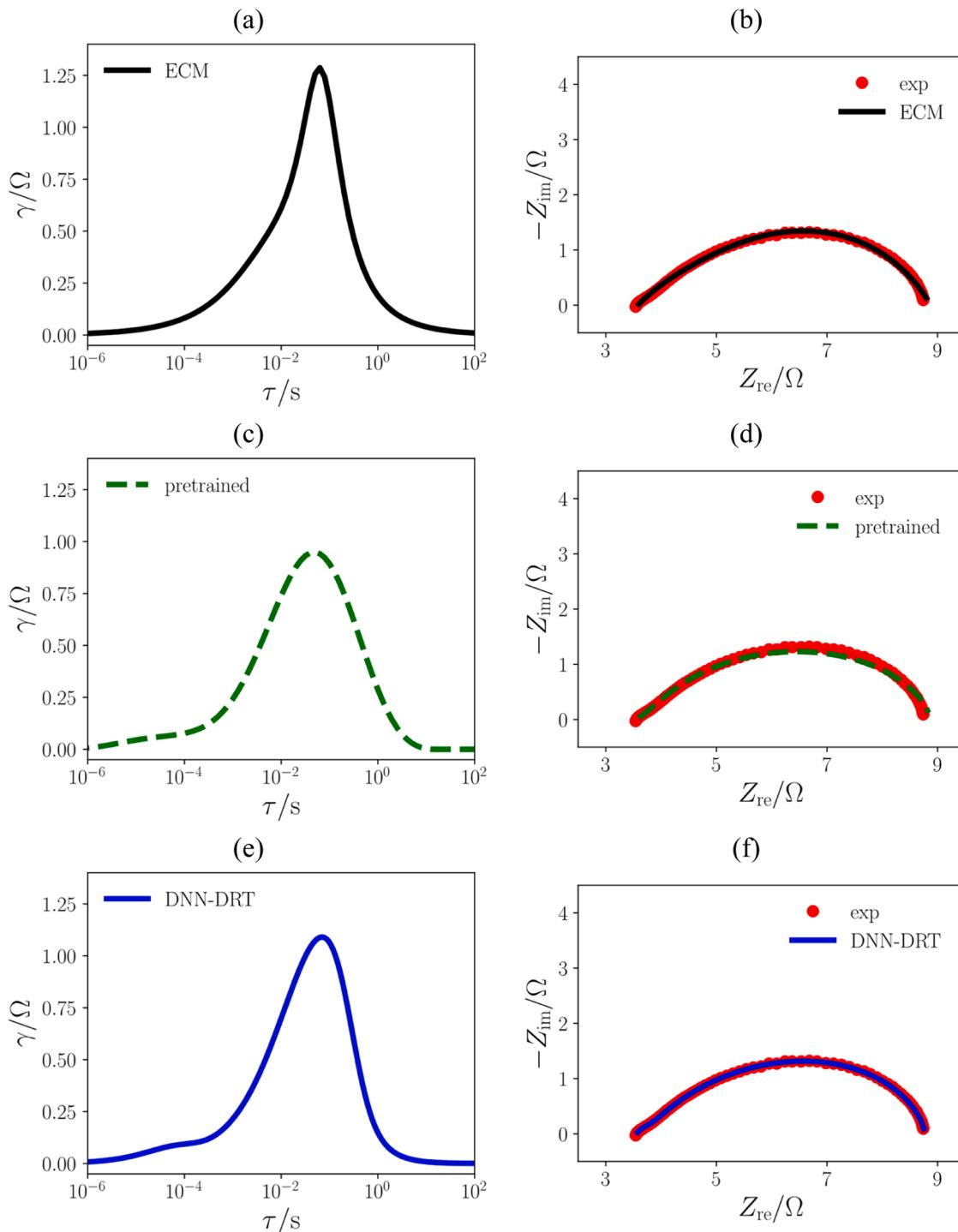


Fig. 9. (a) Fitted-ECM, (c) pretrained DNN, and (e) trained DNN DRTs for the solid oxide fuel cell ($N = 100 M$); corresponding Nyquist plots are reported in panels (b), (d), and (f) with the experimental impedances also shown.

3.2.2. Fuel cells

3.2.2.1 Solid oxide fuel cell. We analyzed the EIS spectrum from a symmetric solid oxide fuel cell with $(\text{Ba}_{0.95}\text{La}_{0.05})_{0.95}\text{FeO}_{3-\delta}$ as the cathode material and samarium-doped ceria as the electrolyte [30,37]. Following the literature [55], we modeled the experimental impedance with a $2 \times \text{ZARC}$ model whose parameters are given in Table S30. After RR, the position of the DRT peak was well identified but the peak height was slightly underfitted (Figs. 9 and S14), which explains the DRT error for RR ($\epsilon_{\gamma,\text{RR}} = 2.07 \times 10^{1\%}$ in Table S28). As discussed in Section 3.2.1,

this is an approximation since an exact value for $\epsilon_{\gamma,\text{RR}}$ cannot be computed. Comparatively, the impedance error for RR was negligible ($\bar{\epsilon}_{Z,\text{RR}} = 9.10 \times 10^{-1\%}$, Table S29). The trained DNN remarkably reduced both γ discrepancy and the error on Z ($\epsilon_{\gamma,\text{DNN}} = 1.33 \times 10^{1\%}$ and $\bar{\epsilon}_{Z,\text{DNN}} = 1.80 \times 10^{-1\%}$ for $N = 10M$).

3.2.2.2 Proton exchange membrane fuel cell. Lastly, we considered the EIS spectrum from a proton exchange membrane fuel cell with platinum on carbon anode and cathode electrodes and a Nafion electrolyte membrane [38]. More details on the assembly and characterization of this

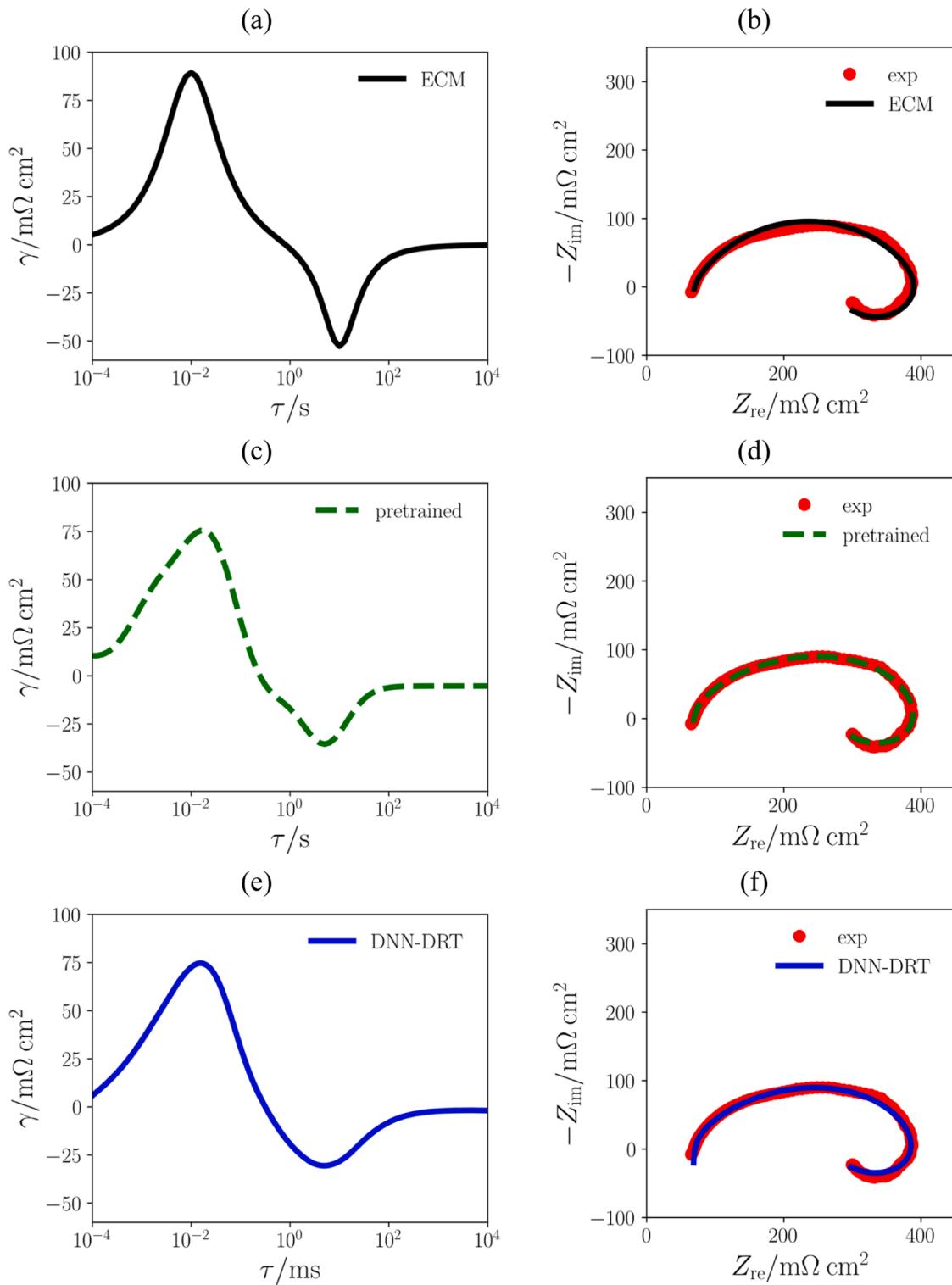


Fig. 10. (a) Fitted-ECM, (c) pretrained DNN, and (e) trained DNN DRTs for the proton exchange membrane fuel cell ($N = 100 \text{ M}$); corresponding Nyquist plots are reported in panels (b), (d), and (f) with the experimental impedances also shown.

fuel cell are provided in Section S5. We modeled this fuel cell with the "hook" model (see Table S31 for the parameters). We proceeded as described in Sections 3.2.1 and 3.2.2.1, and we concluded that the trained DNN closely deconvolved the DRT and fitted the EIS spectrum, see Figs. 10 and S14 for $N = 10 \text{ M}$ and the errors ($\varepsilon_{\gamma,\text{RR}} = 2.97 \times 10^{1\%}$, $\varepsilon_{\gamma,\text{DNN}} = 2.64 \times 10^{1\%}$, $\bar{\varepsilon}_{Z,\text{RR}} = 1.95 \times 10^{0\%}$, and $\bar{\varepsilon}_{Z,\text{DNN}} = 3.19 \times 10^{0\%}$ for $N = 10 \text{ M}$, see Tables S28 and S29).

4. Conclusion

This work develops the DNN-DRT model, which overcomes several shortcomings of DNN-based DRT deconvolution of EIS data by reducing the long pretraining time, assessing the DNN errors accurately, and handling "hook"-type EIS data. The DNN-DRT model performance is assessed against an extensive set of synthetic spectra obtained from

continuous, discontinuous, timescale-overlapping, and negative DRTs. These controlled experiments show that the method is consistent and robust in regressing EIS spectra and deconvolving the DRT. Moreover, in these controlled tests the DNN-DRT model shows greater accuracy than other deconvolution methods, including RR, the deep-DRT, and radial-basis-function approaches. The DNN-DRT method was also tested against real data showing consistency with ECMs. In short, this work develops a holistic approach of effective DRT deconvolution using DNNs by leveraging pretraining, error control, and tuning of DNN architecture and activation functions.

List of Symbols

Greek letters

γ	Vector of discretized DRTs
ε	Vector of experimental errors
θ	Vector of DNN parameters (biases and weights)
τ	Vector of timescales
γ	DRT
ϵ_{integ}	Integration error
ϵ_γ	Normalized DRT error
$\bar{\epsilon}_Z$	Average impedance error with respect to the frequency
ϵ_{trunc}	Truncation error
λ_{RR}	Coefficient of the RR penalty term
σ_n	Noise standard deviation
σ_n^{exp}	Noise standard deviation for the synthetic experiments
τ	Characteristic relaxation timescale

Latin letters

f	Vector of frequencies
I	Identity matrix
Z	Vector of impedances
f	Frequency
L_0	Inductance
$\mathcal{L}_{\text{training}}$	Loss function for DNN training
M	Number of frequencies probed
N	Number of collocation points
P_δ	First penalty term in the training loss function for the "hook" model
P_{cross}	Second penalty term in the training loss function for the "hook" model
R_∞	Ohmic resistance
Z	Impedance

where the subscripts RR, DNN, and exp refer to ridge regression, DNN training, and experimental, respectively.

CRediT authorship contribution statement

Emanuele Quattrocchi: Software, Validation, Methodology, Data curation, Formal analysis, Methodology, Writing – original draft. **Baptiste Py:** Methodology, Investigation, Writing – review & editing. **Adeleke Maradesa:** Investigation, Writing – review & editing. **Quentin Meyer:** Investigation, Writing – review & editing. **Chuan Zhao:** Investigation, Writing – review & editing. **Francesco Ciucci:** Software, Methodology, Validation, Conceptualization, Formal analysis, Methodology, Investigation, Resources, Writing – review & editing, Writing – original draft, Funding acquisition, Project administration, Supervision.

Declaration of Competing Interest

None.

Data availability

Data will be made available on request.

Acknowledgement

The authors gratefully acknowledge the financial support from the Research Grants Council of Hong Kong (RGC Ref. Nos. 16201820 and

16206019). This work was supported in part by the Project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone (HZQB-KCZYB-2020083). The authors would also like to thank HKUST Fok Ying Tung Research Institute and National Supercomputing Center in Guangzhou Nansha sub-center for providing high-performance computational resources. C. Zhao acknowledges the Australian Research Council for financial support (FT170100224, LP200100255, IC200100023). Lastly, E. Quattrocchi, B. Py, and A. Maradesa kindly thank the Hong Kong PhD Fellowship Scheme for its financial support.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.electacta.2022.141499](https://doi.org/10.1016/j.electacta.2022.141499).

References

- [1] P. Iurilli, C. Brivio, V. Wood, On the use of electrochemical impedance spectroscopy to characterize and model the aging phenomena of lithium-ion batteries: a critical review, *J. Power Sources* 505 (2021), 229860.
- [2] Q. Meyer, I. Pivac, F. Barbir, C. Zhao, Detection of oxygen starvation during carbon corrosion in proton exchange membrane fuel cells using low-frequency electrochemical impedance spectroscopy, *J. Power Sources* 470 (2020), 228285.
- [3] E. Quattrocchi, T.H. Wan, A. Curcio, S. Pepe, M.B. Effat, F. Ciucci, A general model for the impedance of batteries and supercapacitors: the non-linear distribution of diffusion times, *Electrochim. Acta* 324 (2019), 134853.
- [4] K. Kruckiewicz, Electrochemical impedance spectroscopy as a versatile tool for the characterization of neural tissue: a mini review, *Electrochim. Commun.* 116 (2020), 106742.
- [5] U.G. Kyle, I. Bosaeus, A.D. De Lorenzo, P. Deurenberg, M. Elia, J.M. Gomez, B. L. Heitmann, L. Kent-Smith, J.C. Melchior, M. Pirlich, H. Scharfetter, A.M. Schols, C. Picard, Bioelectrical impedance analysis-part I: review of principles and methods, *Clin. Nutr.* 23 (2004) 1226–1243.
- [6] A.J. Bard, L.R. Faulkner, J. Leddy, C.G. Zoski, *Electrochemical Methods: Fundamentals and Applications*, Wiley, New York, 1980.
- [7] F. Ciucci, Modeling electrochemical impedance spectroscopy, *Curr. Opin. Electrochim.* 13 (2019) 132–139.
- [8] D.D. Macdonald, Reflections on the history of electrochemical impedance spectroscopy, *Electrochim. Acta* 51 (2006) 1376–1388.
- [9] J. Liu, F. Ciucci, Modeling the impedance spectra of mixed conducting thin films with exposed and embedded current collectors, *Phys. Chem. Chem. Phys.* 19 (2017) 26310–26321.
- [10] A. Lasia, *Electrochemical Impedance Spectroscopy and Its Applications*, Springer, 2014.
- [11] B.A. Boukamp, A. Rolle, R.N. Vannier, R.K. Sharma, E. Djurado, Electrostatic spray deposited $\text{Ca}_3\text{Co}_4\text{O}_{9-\delta}$ and $\text{Ca}_3\text{Co}_4\text{O}_{9-\delta}/\text{Ce}_{0.9}\text{Gd}_{0.1}\text{O}_{1.95}$ cathodes for SOFC, *Electrochim. Acta* 362 (2020), 137142.
- [12] E. Ivers-Tiffée, A. Weber, Evaluation of electrochemical impedance spectra by the distribution of relaxation times, *J. Ceram. Soc. Jpn.* 125 (2017) 193–201.
- [13] F. Ciucci, C. Chen, Analysis of electrochemical impedance spectroscopy data using the distribution of relaxation times: a Bayesian and hierarchical Bayesian approach, *Electrochim. Acta* 167 (2015) 439–454.
- [14] E. Barsoukov, J.R. Macdonald, *Impedance Spectroscopy theory, experiment, and Applications*, 2nd ed., Wiley, 2005.
- [15] R. Franke-Lang, J. Kowal, Analysis of electrochemical impedance spectroscopy on zinc-air batteries using the distribution of relaxation times, *Batteries* 7 (2021) 56.
- [16] T.H. Wan, M. Saccoccia, C. Chen, F. Ciucci, Influence of the discretization methods on the distribution of relaxation times deconvolution: implementing radial basis functions with DRT tools, *Electrochim. Acta* 184 (2015) 483–499.
- [17] M.A. Danzer, Generalized distribution of relaxation times analysis for the characterization of impedance spectra, *Batteries* 5 (2019) 53.
- [18] M. Hahn, S. Schindler, L.C. Trieb, M.A. Danzer, Optimized process parameters for a reproducible distribution of relaxation times analysis of electrochemical systems, *Batteries* 5 (2019) 43.
- [19] N. Schlüter, S. Ernst, U. Schröder, Finding the optimal regularization parameter in distribution of relaxation times analysis, *ChemElectroChem* 6 (2019) 6027–6037.
- [20] R.A. Renaut, R. Baker, M. Horst, C. Johnson, D. Nasir, Stability and error analysis of the polarization estimation inverse problem for microbial fuel cells, *Inverse Probl.* 29 (2013), 045006.
- [21] B.A. Boukamp, Fourier transform distribution function of relaxation times: application and limitations, *Electrochim. Acta* 154 (2015) 35–46.
- [22] K. Kobayashi, T.S. Suzuki, Distribution of relaxation time analysis for non-ideal immittance spectrum: discussion and progress, *J. Phys. Soc. Jpn.* 87 (2018), 094002.
- [23] A.K. Baral, Y. Tsur, Impedance spectroscopy of Gd-doped ceria analyzed by genetic programming (ISGP) method, *Solid State Ion.* 304 (2017) 145–149.
- [24] Q. Wang, Z. Hu, L. Xu, J. Li, Q. Gan, X. Du, M. Ouyang, A comparative study of equivalent circuit model and distribution of relaxation times for fuel cell impedance diagnosis, *Int. J. Energy Res.* 45 (2021) 15948–15961.

- [25] E. Tuncer, S. Gubanski, On dielectric data analysis. Using the Monte Carlo method to obtain relaxation time distribution and comparing non-linear spectral function fits, *IEEE Trans. Dielectr. Electr. Insul.* 8 (2001) 310–320.
- [26] R.E. Koh, C.C. Sun, Y.L. Yap, P.L. Cheang, A.H. You, Investigation of lithium transference number in PMMA composite polymer electrolytes using Monte Carlo (MC) simulation and recurrence relation, *J. Electrochem. Sci. Technol.* 12 (2021) 217–224.
- [27] F. Dion, A. Lasia, The use of regularization methods in the deconvolution of underlying distributions in electrochemical processes, *J. Electroanal. Chem.* 475 (1999) 28–37.
- [28] M. Saccoccia, T.H. Wan, C. Chen, F. Ciucci, Optimal regularization in distribution of relaxation times applied to electrochemical impedance spectroscopy: ridge and Lasso regression methods - a theoretical and experimental study, *Electrochim. Acta* 147 (2014) 470–482.
- [29] J. Liu, F. Ciucci, The deep-prior distribution of relaxation times, *J. Electrochem. Soc.* 167 (2020), 026506.
- [30] E. Quattrochi, T.H. Wan, A. Belotti, D. Kim, S. Pepe, S.V. Kalinin, M. Ahmadi, F. Ciucci, The deep-DRT: a deep neural network approach to deconvolve the distribution of relaxation times from multidimensional electrochemical impedance spectroscopy data, *Electrochim. Acta* 392 (2021), 139010.
- [31] X. Li, M. Ahmadi, L. Collins, S.V. Kalinin, Deconvolving distribution of relaxation times, resistances and inductance from electrochemical impedance spectroscopy via statistical model selection: exploiting structural-sparsity regularization and data-driven parameter tuning, *Electrochim. Acta* 313 (2019) 570–583.
- [32] M. Belkin, D. Hsu, S. Ma, S. Mandal, Reconciling modern machine-learning practice and the classical bias-variance trade-off, *Proc. Natl Acad. Sci.* 116 (2019) 15849–15854.
- [33] C.C. Chen, C.L. Yang, H.Y. Cheng, Efficient and robust parallel dnn training through model parallelism on multi-gpu platform, arXiv preprint arXiv: 1809.02839, (2018). doi: [10.48550/arXiv.1809.02839](https://doi.org/10.48550/arXiv.1809.02839).
- [34] D. Choi, H. Cho, W. Rhee, On the difficulty of DNN hyperparameter optimization using learning curve prediction, in: Proceedings of the TENCON 2018-2018 IEEE Region 10 Conference, IEEE, 2018, pp. 0651–0656.
- [35] D. Erhan, A. Courville, Y. Bengio, P. Vincent, Why does unsupervised pre-training help deep learning?, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.
- [36] J. Yu, Y.Q. Lyu, J. Liu, M.B. Effat, S.C.T. Kwok, J. Wu, F. Ciucci, Enabling non-flammable Li-metal batteries via electrolyte functionalization and interface engineering, *J. Mater. Chem. A* 7 (2019) 17995–18002.
- [37] A. Belotti, Y. Wang, A. Curcio, J. Liu, E. Quattrochi, S. Pepe, F. Ciucci, The influence of A-site deficiency on the electrochemical properties of $(\text{Ba}_{0.95}\text{La}_{0.05})_{1-x}\text{FeO}_{3-\delta}$ as an intermediate temperature solid oxide fuel cell cathode, *Int. J. Hydrog. Energy* 47 (2) (2021) 1229–1240.
- [38] Q. Meyer, C. Zhao, Air perturbation-induced low-frequency inductive electrochemical impedance arc in proton exchange membrane fuel cells, *J. Power Sources* 488 (2021), 229245.
- [39] D. Calvetti, E. Somersalo, Computational Mathematical modeling: an Integrated Approach Across Scales, SIAM, 2012.
- [40] S. Sankararaman, S. Mahadevan, Model validation under epistemic uncertainty, *Reliab. Eng. Syst. Saf.* 96 (2011) 1232–1241.
- [41] M.E. Orazem, B. Tribollet, *Electrochemical Impedance Spectroscopy*, Wiley, New Jersey, 2008, pp. 383–389.
- [42] D. Klotz, Negative capacitance or inductive loop? A general assessment of a common low frequency impedance feature, *Electrochim. Commun.* 98 (2019) 58–62.
- [43] S. Diamond, S. Boyd, CVXPY: a python-embedded modeling language for convex optimization, *J. Mach. Learn. Res.* 17 (2016) 2909–2913.
- [44] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980, (2014).
- [45] D. Maclaurin, D. Duvenaud, R.P. Adams, Autograd: effortless gradients in numpy, in: Proceedings of the ICML 2015 AutoML Workshop, 2015, p. 5.
- [46] G. Parascandolo, H. Huttunen, T. Virtanen, Taming the waves: sine as activation function in deep neural networks, in: Proceedings of the ICLR 2017, 2016.
- [47] H. Zheng, Z. Yang, W. Liu, J. Liang, Y. Li, Improving deep neural networks using softplus units, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, 2015, pp. 1–4.
- [48] D.A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), arXiv preprint arXiv:1511.07289, (2015).
- [49] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
- [50] in: X. Glorot, Y. Bengio, T. Yee Whye, T. Mike, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR, Proceedings of Machine Learning Research, 2010, pp. 249–256.
- [51] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic Differentiation in PyTorch, 2017, NIPS, 2017.
- [52] E. Barsoukov, *Impedance Spectroscopy: Theory, Experiment, and Applications*, John Wiley, 2017.
- [53] B.A. Boukamp, A. Rolle, Analysis and application of distribution of relaxation times in solid state ionics, *Solid State Ion.* 302 (2017) 12–18.
- [54] N. Schlüter, S. Ernst, U. Schröder, Direct access to the optimal regularization parameter in distribution of relaxation times analysis, *ChemElectroChem* 7 (2020) 3445–3458.
- [55] A. Jun, J. Kim, J. Shin, G. Kim, Perovskite as a cathode material: a review of its role in solid-oxide fuel cell technology, *ChemElectroChem* 3 (2016) 511–530.