Deconvolution of Electrochemical Impedance Spectroscopy Data Using the Deep-Neural-Network-Enhanced Distribution of Relaxation Times

Emanuele Quattrocchi[a], Baptiste Py[a], Adeleke Maradesa[a], Quentin Meyer[b], Chuan Zhao[b], Francesco Ciucci[a,c,d,e*]

[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] Department of Chemical and Biological Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China

[d] HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen 51048, China

[e] Energy Institute, The Hong Kong University of Science and Technology, Hong Kong SAR, China

[*] Corresponding author: francesco.ciucci@ust.hk

Phone: +852 2358 7187

Fax: +852 2358 1543

Abstract

Electrochemical impedance spectroscopy (EIS) is a characterization technique widely used to evaluate the properties of electrochemical systems. The distribution of relaxation times (DRT) has emerged as a model-free alternative to equivalent circuits and physical models to circumvent the inherent challenges of EIS analysis. Deep neural networks (DNNs) can be used to deconvolve DRTs, but several issues remain, *e.g.*, the long training time, the DNN accuracy, and the deconvolution of DRTs with negative peaks. The DNN-DRT model was developed here to address these fundamental limitations. Specifically, a pretraining step was included to decrease the computation time. A thorough error analysis was also conducted to evaluate the different components of the DRT and impedance errors to ultimately decrease them. Lastly, the training loss function was modified to handle DRTs with negative peaks. These different advances were validated with an array of synthetic EIS spectra and real EIS spectra from a lithium-ion battery, a solid oxide fuel cell, and a proton exchange membrane fuel cell. Moreover, this new model outperformed in most cases the previously developed DRTtools and deep-DRT model. Overall, we envision that this research will open the venue for more DNN-based analyses of EIS data for electrochemical systems.

Keywords: Electrochemical impedance spectroscopy, Distribution of relaxation times, Neural networks, Error analysis, Lithium-ion batteries, Fuel cells

1 Introduction

Electrochemical impedance spectroscopy (EIS) is an experimental technique widely used for the analysis of electrochemical energy devices (*e.g.*, batteries [1-5], fuel cells [6-10], and supercapacitors [11-14]), medical [15, 16], and biological [17-19] systems. The major appeals of this technique are that it can be carried out over a wide range of frequencies (from mHz to MHz) [20] and that modern equipment allows us to measure EIS spectra quickly and efficiently with limited user expertise. To interpret EIS data, equivalent circuits and physical models are often utilized [21-23]. However, 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 [24, 25]. Considering these challenges, the distribution of relaxation times (DRT) has recently emerged as an alternative approach to analyze EIS data [26-29]. The DRT model assumes that the electrochemical processes underpinning EIS spectra are relaxations [30-34]. This means that the DRT impedance model, $Z_{\mathrm{DRT}}(f)$, at a given frequency, f, is obtained using the following expression [35, 36]

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

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 [5, 37]. Despite the DRT's considerable promise, deconvolving $\gamma(\log\tau)$ from experimental data is still challenging. Indeed, the inverse problem that needs to be solved to obtain $\gamma(\log\tau)$ is ill-posed with a strong dependence on experimental errors and cutoff frequencies [29, 38, 39]. To estimate $\gamma(\log\tau)$, existing methods leverage Fourier transform [40, 41], genetic algorithms [42, 43], Monte Carlo samplers [44, 45], and ridge regression (RR) [37, 46-49]. In particular, RR seeks to minimize the following loss function [37, 47]:

$$\mathcal{L}(\boldsymbol{\gamma}) = \left\| \boldsymbol{Z}_{\mathrm{exp}}(\boldsymbol{f}) - \boldsymbol{Z}_{\mathrm{DRT}}(\boldsymbol{f}, \boldsymbol{\gamma}) \right\|^2 + P(\boldsymbol{\gamma}) \tag{2}$$

where $\boldsymbol{\gamma}$, $\boldsymbol{Z}_{\mathrm{exp}}$, $\boldsymbol{f}$, and $\boldsymbol{Z}_{\mathrm{DRT}}$ are the vectors of discretized DRTs, experimental impedances, experimental frequencies, and DRT impedances, respectively, $\|\cdot\|$ is the Euclidean 2-norm, and $P(\boldsymbol{\gamma})$ is a penalty function.

It was recently shown that deep neural networks (DNNs) can be utilized to deconvolve the DRT [50, 51]. In particular, the first DNN-based method ever developed, called deep-prior DRT, used a DNN with a single random scalar as input [50] and output a vector of DRT values at discrete 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 for the DNN. Unlike in conventional RR, regularization was not necessary to obtain $\gamma(\log\tau)$ [47, 52, 53]. Another particularly interesting aspect of the deep-prior DRT approach is that the corresponding inverse problem is overparametrized: 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 introduced by Belkin *et al.* [54, 55], overparameterization allows for small test errors, thereby improving the DNN performance. Quattrocchi *et al.* further extended the deep-prior DRT approach with the deep-DRT model by considering a DNN that takes as inputs the scalar log-timescale $\log\tau$ and a state vector $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_{\mathrm{P}})$ describing experimental conditions (*e.g.*, temperature, pressure, etc.) [51]. 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 not tested experimentally.

While the literature highlights the considerable potential of DNN-based methods for DRT deconvolution, several shortcomings need to be resolved for enhanced EIS analyses. i) DNN

training requires several tens of thousands of iterations, which is time consuming [56, 57]. ii) Even though the DNN is a universal approximator that closely matches any function (see Section S1 of the supplementary information (SI)), DRT deconvolution has been performed with a limited accuracy because the integral in (1) needs to be approximated at a fixed number of collocation points [47, 50]. iii) The recovery of DRTs with negative peaks has yet to be studied as the output of DNN methods that have been constrained to be non-negative by means of a dedicated activation function at the DNN's last layer [50, 51]. This article aims to address thesechallenges.

To reduce the computational time, the DNN-DRT model developed herein includes a pretraining step. As shown in Figure 1, RR is used to obtain a coarse DRT estimate and pretrain the DNN [58]. Then, the accuracy of the pretrained DNN is improved by minimizing a loss function generally without penalty (*i.e.*, (2) without the $P(\boldsymbol{\gamma})$ term). Second, postprocessing is used to analyze errors, and, if needed, the obtained model error is minimized by refining the grid over which the integral in (1) is computed. By removing the non-negativity constraint and introducing suitable penalty terms, the DNN-DRT model can 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-ion battery [59], a solid oxide fuel cell [60], and a proton exchange membrane fuel cell [61].

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 for energy systems and beyond.

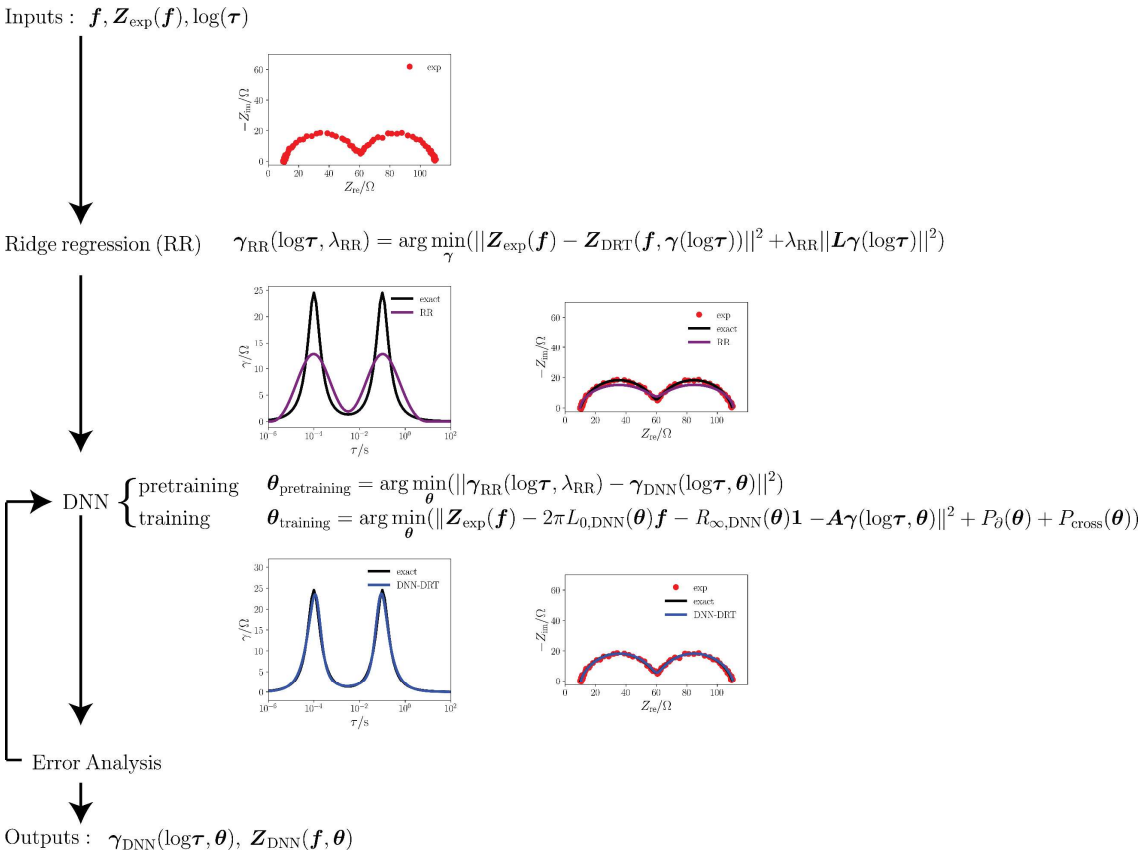

Figure 1: First, RR is conducted to derive a coarse DRT $\boldsymbol{\gamma}_{\mathrm{RR}}$, which is then used for pretraining to obtain initial values $\boldsymbol{\theta}_{\mathrm{pretraining}}$ of the DNN parameters used during training to optimize $\boldsymbol{\gamma}_{\mathrm{DNN}}(\log\tau, \boldsymbol{\theta})$. Lastly, an error analysis is conducted to increase the DNN accuracy.

2 Methods

2.1 The DNN-DRT Model

The DRT is encoded through a DNN and is denoted as $\gamma_{\mathrm{DNN}}(\log\tau, \boldsymbol{\theta})$ with $\boldsymbol{\theta}$ 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 $\log\boldsymbol{\tau} = (\log\tau_1, \log\tau_2, \ldots, \log\tau_N)^{\mathrm{T}}$ where $\tau_1 = \tau_{\min}$ and $\tau_N = \tau_{\max}$ with $\tau_1 < \tau_2 < \cdots < \tau_N$.

Therefore, using a midpoint rule to compute (1), the impedance $Z_{\mathrm{DNN}}(f,\boldsymbol{\theta})$ at a frequency f is given by [47]

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

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

Given the vector of experimental impedances, $\boldsymbol{Z}_{\mathrm{exp}}(\boldsymbol{f})$, probed at the M frequencies $\boldsymbol{f} = (f_1, f_2, \dots, f_M)^\top$, we aim to obtain a latent DRT vector $\boldsymbol{\gamma}_{\mathrm{DNN}}(\log\boldsymbol{\tau},\boldsymbol{\theta}) = \left(\gamma_{\mathrm{DNN}}(\log\tau_1,\boldsymbol{\theta}),\gamma_{\mathrm{DNN}}(\log\tau_2,\boldsymbol{\theta}),\dots,\gamma_{\mathrm{DNN}}(\log\tau_N,\boldsymbol{\theta})\right)^\top$ at the vector of log-timescales $\log\boldsymbol{\tau}$. We will assume the following discretized version of (1) [47]:

$$\boldsymbol{Z}_{\mathrm{DNN}}(\boldsymbol{f},\boldsymbol{\theta}) = i2\pi L_0\boldsymbol{f} + R_\infty\mathbf{1} + (\mathbf{A}_{\mathrm{re}} + i\mathbf{A}_{\mathrm{im}})\boldsymbol{\gamma}_{\mathrm{DNN}}(\log\boldsymbol{\tau},\boldsymbol{\theta}) \tag{4}$$

where the matrices $\mathbf{A}_{\mathrm{re}}$ and $\mathbf{A}_{\mathrm{im}} \in \mathbb{R}^{M\times N}$ are given in the literature [47], $\mathbf{1} = (1,1,\dots,1)^\top \in \mathbb{R}^M$, and $\boldsymbol{Z}_{\mathrm{DNN}}(\boldsymbol{f},\boldsymbol{\theta}) = \left(Z_{\mathrm{DNN}}(f_1,\boldsymbol{\theta}),Z_{\mathrm{DNN}}(f_2,\boldsymbol{\theta})\dots,Z_{\mathrm{DNN}}(f_M,\boldsymbol{\theta})\right)^\top$ is the vector of DNN impedances.

2.2 Errors

It is assumed that the experimental impedance $\boldsymbol{Z}_{\mathrm{exp}}(\boldsymbol{f}) = \left(Z_{\mathrm{exp}}(f_1),Z_{\mathrm{exp}}(f_2),\dots,Z_{\mathrm{exp}}(f_M)\right)^\top$ is given by

$$\boldsymbol{Z}_{\mathrm{exp}}(\boldsymbol{f}) = \boldsymbol{Z}_{\mathrm{DNN}}(\boldsymbol{f},\boldsymbol{\theta}) + \boldsymbol{\varepsilon}_{\mathrm{epist}} + \boldsymbol{\varepsilon}_{\mathrm{trunc}} + \boldsymbol{\varepsilon}_{\mathrm{integ}} + \boldsymbol{\varepsilon}_{\mathrm{meas}} \tag{5}$$

where $\boldsymbol{\varepsilon}_{\text{epist}}$, $\boldsymbol{\varepsilon}_{\text{trunc}}$, $\boldsymbol{\varepsilon}_{\text{integ}}$, and $\boldsymbol{\varepsilon}_{\text{meas}}$ are the vectors of epistemic, truncation, integration, and measurement errors, respectively [39, 62]. In this article, $\boldsymbol{\varepsilon}_{\text{epist}}$ is not considered. The errors due to the numerical methods include truncation, $\boldsymbol{\varepsilon}_{\text{trunc}}$, and integration, $\boldsymbol{\varepsilon}_{\text{integ}}$ (see Section 2.4). Finally, EIS data is inherently laden with experimental noise, $\boldsymbol{\varepsilon}_{\text{meas}}$. Hereon, $\boldsymbol{\varepsilon}_{\text{meas}}$ is assumed to be normally distributed, *i.e.*, $\boldsymbol{\varepsilon}_{\text{meas}} \sim \mathcal{N}(0, \sigma_n^2\mathbf{I})$ (σ_n is a scalar and $\mathbf{I}$ the $M \times M$ identity matrix) [37, 47, 51].

2.3 DRT Model

2.3.1 Ridge Regression

If RR is used to obtain the DRT, then the vector of the discretized DRT from RR, $\boldsymbol{\gamma}_{\text{RR}}(\log\boldsymbol{\tau}, \lambda_{\text{RR}}) = \left(\gamma_{\text{RR}}(\log\tau_1, \lambda_{\text{RR}}), \gamma_{\text{RR}}(\log\tau_2, \lambda_{\text{RR}}), \ldots, \gamma_{\text{RR}}(\log\tau_N, \lambda_{\text{RR}})\right)^{\mathsf{T}}$, is the solution of the following problem:

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

where $\boldsymbol{L}$ is a differentiation matrix and λ_{RR} is a scalar [47].

2.3.2 DNN deconvolution

To obtain $\boldsymbol{\gamma}_{\text{DNN}}(\log\boldsymbol{\tau}, \boldsymbol{\theta})$, pretraining is carried out first (see Figure 1). In particular, the weights and biases are optimized as

$$\boldsymbol{\theta}_{\text{pretraining}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\|\boldsymbol{\gamma}_{\text{RR}}(\log\boldsymbol{\tau}, \lambda_{\text{RR}}) - \boldsymbol{\gamma}_{\text{DNN}}(\log\boldsymbol{\tau}, \boldsymbol{\theta})\|^2 \tag{7}$$

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

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

with respect to $\boldsymbol{\theta}$ where $\boldsymbol{\theta}_{\text{pretraining}}$ is taken as the starting point of the minimization (see Figure 1). We must stress that $\boldsymbol{\gamma}_{\text{DNN}}(\log \boldsymbol{\tau}, \boldsymbol{\theta})$ in (7) must be discretized at the same collocation points as $\boldsymbol{\gamma}_{\text{RR}}(\log \boldsymbol{\tau}, \lambda_{\text{RR}})$. Nonetheless, the log-timescale vectors $\log \boldsymbol{\tau}$ in (7) and in (8) are not necessarily identical since a denser grid for $\boldsymbol{\gamma}_{\text{DNN}}(\log \boldsymbol{\tau}, \boldsymbol{\theta})$ can be used during training (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" [61, 63], have DRTs with negative peaks. To account for this, the non-negativity constraint in (6) was removed. Furthermore, two penalty terms $P_{\partial}(\boldsymbol{\theta})$ and $P_{\text{cross}}(\boldsymbol{\theta})$ were added to (8) for DNN training, leading to the following loss function to be minimized

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

where $P_{\partial}(\boldsymbol{\theta})$ is defined as

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

where $q \in \mathbb{Z}^+$ and $\lambda(\log \tau, \boldsymbol{\theta})$ is a coefficient estimated in a hyper Bayesian framework leveraging the RR parameter λ_{RR}, see Section S4 [30].

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

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

with $\log\tau_n$ the n^{th} log-timescale for $n = 1, 2, \ldots, N$, and α_1, α_2, β_1, and β_2 four scalars defined as $\alpha_1 = \frac{\log\left(\frac{\tau_N}{\tau_1}\right)}{2(N-1)}$, $\alpha_2 = \frac{\alpha_1}{2}$, $\beta_1 = \frac{1}{\alpha_1}$, and $\beta_2 = 2\beta_1$ (see Section S3.2).

2.4 Error Indicators

To evaluate the quality of the DRT deconvolution and impedance recovery, various error metrics were used [51], namely the normalized DRT error for RR $\varepsilon_{\gamma,\text{RR}}$ and for DNN training $\varepsilon_{\gamma,\text{DNN}}$, defined as:

$$\varepsilon_{\gamma,\text{RR}} = \frac{\|\boldsymbol{\gamma}_{\text{exact}} - \boldsymbol{\gamma}_{\text{RR}}\|}{\|\boldsymbol{\gamma}_{\text{exact}}\|} \tag{12a}$$

$$\varepsilon_{\gamma,\text{DNN}} = \frac{\|\boldsymbol{\gamma}_{\text{exact}} - \boldsymbol{\gamma}_{\text{DNN}}\|}{\|\boldsymbol{\gamma}_{\text{exact}}\|} \tag{12b}$$

where $\boldsymbol{\gamma}_{\text{exact}}(\log\boldsymbol{\tau}) = \left(\gamma_{\text{exact}}(\log\tau_1), \gamma_{\text{exact}}(\log\tau_2), \ldots, \gamma_{\text{exact}}(\log\tau_N)\right)^{\mathsf{T}}$ is the discretized DRT vector computed with the analytical formulas given in Section 2.5.5. We must stress that $\boldsymbol{\gamma}_{\text{exact}}(\log\boldsymbol{\tau})$ was discretized on the same grid as $\boldsymbol{\gamma}_{\text{RR}}(\log\boldsymbol{\tau}, \lambda_{\text{RR}})$ in (12a) and as $\boldsymbol{\gamma}_{\text{DNN}}(\log\boldsymbol{\tau}, \boldsymbol{\theta})$ in (12b).

The quality of the impedance recovery was assessed with four frequency-dependent indicators, namely the truncation $\varepsilon_{\text{trunc}}(f)$, integration $\varepsilon_{\text{integ}}(f)$, RR $\varepsilon_{\text{Z,RR}}(f)$, and DNN training $\varepsilon_{\text{Z,DNN}}(f)$ errors, whose definitions are given in Table 1. The expressions of the truncation errors on the real and imaginary parts of the impedance, $\varepsilon_{\text{trunc,re}}(f)$ and $\varepsilon_{\text{trunc,im}}(f)$, and those of the integration errors, $\varepsilon_{\text{integ,re}}(f)$ and $\varepsilon_{\text{nteg,im}}(f)$, are derived in Section S2 and S3.

Additionally, the frequency-averaged $\bar{\varepsilon}_{\mathrm{trunc}}$, $\bar{\varepsilon}_{\mathrm{integ}}$, $\bar{\varepsilon}_{\mathrm{Z,RR}}$, and $\bar{\varepsilon}_{\mathrm{Z,DNN}}$ of $\varepsilon_{\mathrm{trunc}}(f)$, $\varepsilon_{\mathrm{integ}}(f)$, $\varepsilon_{\mathrm{Z,RR}}(f)$, and $\varepsilon_{\mathrm{Z,DNN}}(f)$, respectively, are presented in Table 1.

Table 1 – Truncation, integration, RR, and DNN errors, and their corresponding frequency averages.

Error type	Frequency-dependent	Frequency-averaged
Truncation	$\varepsilon_{\mathrm{trunc}}(f) = \frac{\left\lvert \varepsilon_{\mathrm{trunc,re}}(f) + i\varepsilon_{\mathrm{trunc,im}}(f) \right\rvert}{\left\lvert Z_{\mathrm{exp}}(f) \right\rvert}$	$\bar{\varepsilon}_{\mathrm{trunc}} = \frac{1}{M}\sum_{m=1}^{M} \varepsilon_{\mathrm{trunc}}(f_m)$
Integration	$\varepsilon_{\mathrm{integ}}(f) = \frac{\left\lvert \varepsilon_{\mathrm{integ,re}}(f) + i\varepsilon_{\mathrm{integ,im}}(f) \right\rvert}{\left\lvert Z_{\mathrm{exp}}(f) \right\rvert}$	$\bar{\varepsilon}_{\mathrm{integ}} = \frac{1}{M}\sum_{m=1}^{M} \varepsilon_{\mathrm{integ}}(f_m)$
RR	$\varepsilon_{\mathrm{Z,RR}}(f) = \frac{\left\lvert Z_{\mathrm{exp}}(f) - Z_{\mathrm{RR}}(f,\boldsymbol{\theta}) \right\rvert}{\left\lvert Z_{\mathrm{exp}}(f) \right\rvert}$	$\bar{\varepsilon}_{\mathrm{Z,RR}} = \frac{1}{M}\sum_{m=1}^{M} \varepsilon_{\mathrm{Z,RR}}(f_m)$
DNN	$\varepsilon_{\mathrm{Z,DNN}}(f) = \frac{\left\lvert Z_{\mathrm{exp}}(f) - Z_{\mathrm{DNN}}(f,\boldsymbol{\theta}) \right\rvert}{\left\lvert Z_{\mathrm{exp}}(f) \right\rvert}$	$\bar{\varepsilon}_{\mathrm{Z,DNN}} = \frac{1}{M}\sum_{m=1}^{M} \varepsilon_{\mathrm{Z,DNN}}(f_m)$

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 $k = 1, \ldots, M$ $(N = M)$, which is common practice [5, 37, 47]. The penalty coefficient λ_{RR} was taken between 10^{-6}, …, and 10^{-1} (the values used are given in Table S1). The problem (6) was solved using CVXPY [64].

2.5.2 DNN Pretraining

To deconvolve the DRT, the problem in (7) was first normalized with respect to the RR polarization resistance, R_{p}, obtained by integrating $\gamma_{\mathrm{RR}}(\log\tau, \lambda_{\mathrm{RR}})$ from $\log\tau_{\mathrm{min}}$ to $\log\tau_{\mathrm{max}}$ with the trapezoidal rule. Then, the optimization in (7) was carried out with Adam algorithm for 2,000 steps [65].

2.5.3 DNN Training

The DNN parameters $\boldsymbol{\theta}$ were obtained by minimizing the loss function in (8) (see Figure 1). The penalty terms, $P_{\partial}(\boldsymbol{\theta})$ and $P_{\text{cross}}(\boldsymbol{\theta})$, were added to the loss function $\mathcal{L}_{\text{training}}(\boldsymbol{\theta})$ in (9) only for EIS spectra with inductive features (Section 2.3.3). For $P_{\partial}(\boldsymbol{\theta})$, we set $q = 1$ in (10), and we computed the first derivative of $\gamma_{\text{DNN}}(\log\tau, \boldsymbol{\theta})$ by backward differentiation [66]. The influence of the grid density on the DRT recovery was evaluated by discretizing $\gamma_{\text{DNN}}(\log\tau, \boldsymbol{\theta})$ with $N = M$, 10 M, and 100 M. The error metrics $\varepsilon_{\gamma,\text{RR}}$ and $\varepsilon_{\gamma,\text{DNN}}$ in (12a) and in (12b) were computed in the range $[\log\tau_1, \log\tau_N]$ utilizing the trapezoidal rule with the grid density used in RR and DNN training, respectively.

2.5.4 Deep Neural Network Architecture and Optimization

For the simulations, we used a vanilla feedforward DNN consisting of 12 layers with 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; and iii) an output layer of dimension 1 that outputs $\gamma_{\text{DNN}}(\log\tau, \boldsymbol{\theta})$. The first layer had sinusoidal activation function [67]. If the $\gamma(\log\tau) \geq 0$ constraint was enforced, the hidden and output layers had softplus activation functions [68]. When the $\gamma(\log\tau) \geq 0$ constraint was relaxed, non-saturating exponential unit functions were used for the hidden and output layers [69]. The DNN had 10,657 parameters. L_0 and R_∞ were also optimized during training. For DNN pretraining, the weights were initialized using the Xavier method (normal or uniform), and the biases set to zero or sampled from a normal distribution (Table S2). Then, the weights and biases of the pretrained DNN were used as initial values for DNN training. The learning rates for pretraining and training were set to 10^{-3} and 10^{-4}, respectively. To avoid overfitting, the early stopping threshold was set to 10^{-8} between two consecutive evaluations of the loss, and the optimal iteration corresponded to the minimum value of $\varepsilon_{\gamma,\text{DNN}}$. The model was implemented with PyTorch [70].

2.5.5 Generation of the Synthetic Experiments

Synthetic spectra were generated using the analytical DRTs and impedances reported in Table 2. We used $M = 81$ log-spaced frequencies between 10^{-2} and 10^{6} Hz with ten points per decade (ppd). Unless otherwise specified, the random errors, $\boldsymbol{\varepsilon}_{\mathrm{meas}}$, were computed according to Section 2.2 with $\sigma_n^{\exp} = 0.5\ \Omega$.

Table 2 – Analytical DRTs and impedances used in the synthetic experiments where R_{ct}, $R_{\mathrm{ct},1}$, $R_{\mathrm{ct},2}$ 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.

Model	$\gamma_{\mathrm{exact}}(\log\tau)$	$Z_{\mathrm{exact}}(f)$	Referenc
Single ZARC	$\frac{R_{\mathrm{ct}}}{2\pi}\frac{\sin\left((1-\phi)\pi\right)}{\cosh\left(\phi\log\left(\frac{\tau}{\tau_0}\right)\right)-\cos\left((1-\phi)\pi\right)}$	$R_\infty+\frac{R_{\mathrm{ct}}}{1+(i2\pi f\tau_0)^{\phi}}$	[25]
2×ZARC and Hook	$\frac{R_{\mathrm{ct},1}}{2\pi}\frac{\sin\left((1-\phi_1)\pi\right)}{\cosh\left(\phi_1\log\left(\frac{\tau}{\tau_1}\right)\right)-\cos\left((1-\phi_1)\pi\right)}+\frac{R_{\mathrm{ct},2}}{2\pi}\frac{\sin\left((1-\phi_2)\pi\right)}{\cosh\left(\phi_2\log\left(\frac{\tau}{\tau_2}\right)\right)-\cos\left((1-\phi_2)\pi\right)}$	$R_\infty+\frac{R_{\mathrm{ct},1}}{1+(i2\pi f\tau_1)^{\phi_1}}+\frac{R_{\mathrm{ct},2}}{1+(i2\pi f\tau_2)^{\phi_2}}$	[25]
PWC	$\frac{R_{\mathrm{ct}}\left(H(\tau-\tau_0)-H(\tau-\tau_1)\right)}{\log\left(\frac{\tau_1}{\tau_0}\right)}$	$R_\infty+\frac{R_{\mathrm{ct}}}{\log\left(\frac{\tau_1}{\tau_0}\right)}\left(\log\left(1-\frac{i}{2\pi f\tau_0}\right)-\log\left(1-\frac{i}{2\pi f\tau_1}\right)\right)$	[30]
Fractal	$\begin{cases}\frac{R_{\mathrm{ct}}}{\pi}\sin(\phi\pi)\left(\frac{\tau}{\tau_0-\tau}\right)^{\phi} & \text{if } \tau<\tau_0\\ 0 & \text{otherwise}\end{cases}$	$R_\infty+\frac{R_{\mathrm{ct}}}{(1+i2\pi f\tau_0)^{\phi}}$	[71]
Gerischer	$\begin{cases}\frac{R_{\mathrm{ct}}}{\pi}\sqrt{\frac{\tau}{\tau_0-\tau}} & \text{if } \tau<\tau_0\\ 0 & \text{otherwise}\end{cases}$	$R_\infty+\frac{R_{\mathrm{ct}}}{\sqrt{1+i2\pi f\tau_0}}$	[31]

3 Results

The DNN-DRT model was firstly benchmarked against synthetic experiments[1]. Using the single ZARC model, we sought to evaluate the magnitude of the truncation $\varepsilon_{\mathrm{trunc}}$ and integration $\varepsilon_{\mathrm{integ}}$ errors. We also studied the influence of the number of collocation points, N, and of the experimental noise, $\boldsymbol{\varepsilon}_{\mathrm{meas}}$, on the DRT recovery. Next, we considered more complex models with overlapping features ($2\times$ZARC), discontinuities (fractal, piecewise constant (PWC), Gerischer), and inductive features (hook). Finally, we tested the DNN-DRT model 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). Figure 2 (a) shows the exact, RR, and pretrained DNN DRTs for $N = 10\ M$. Graphically, we observed that RR and pretraining enabled us to identify the DRT peak. Figure 2 (b) displays the exact, pretrained DNN, and trained DNN DRTs. The DNN DRT recovery was greatly improved compared to the RR DRT, which was confirmed by the normalized DRT errors for RR ($\varepsilon_{\gamma,\mathrm{RR}} = 42.25\%$, see Table S4) and for DNN training ($\varepsilon_{\gamma,\mathrm{DNN}} \approx 5.50\%$). Figure 2 (c) shows $\mathcal{L}_{\mathrm{training}}(\boldsymbol{\theta})$ (see (8)) and $\varepsilon_{\gamma,\mathrm{DNN}}$ (see (12b)) as functions of the iteration number. Both reached their minimum after about 5,000 iterations, while it took approximately 8,000 iterations without pretraining, see Figure S1. After that minimum was reached, $\mathcal{L}_{\mathrm{training}}(\boldsymbol{\theta})$ oscillated around its minimum value while $\varepsilon_{\gamma,\mathrm{DNN}}$ gradually increased. Figure 2 (d) displays the experimental and exact

[1] The values of λ_{RR} used for RR (see (6)) are summarized in Table S1.

impedances, and Figure 2 (e) additionally includes the RR and pretrained DNN impedances. Pretraining the DNN enabled to match the RR impedance. Moreover, Figure 2 (f) shows the experimental, exact, pretrained DNN, and trained DNN impedances. We found that the DNN training enabled us to match the experimental impedance more closely, which was confirmed by the averaged impedance errors for RR ($\bar{\varepsilon}_{\mathrm{Z,RR}} = 5.69\%$, Table S5) and for DNN training ($\bar{\varepsilon}_{\mathrm{Z,DNN}} \approx 3.40\%$). Taking the averages of $\bar{\varepsilon}_{\mathrm{Z,RR}}$ and $\bar{\varepsilon}_{\mathrm{Z,DNN}}$ from ten synthetic spectra led to similar values (Tables S6 and S7), which confirmed that DNN training improved the impedance recovery.

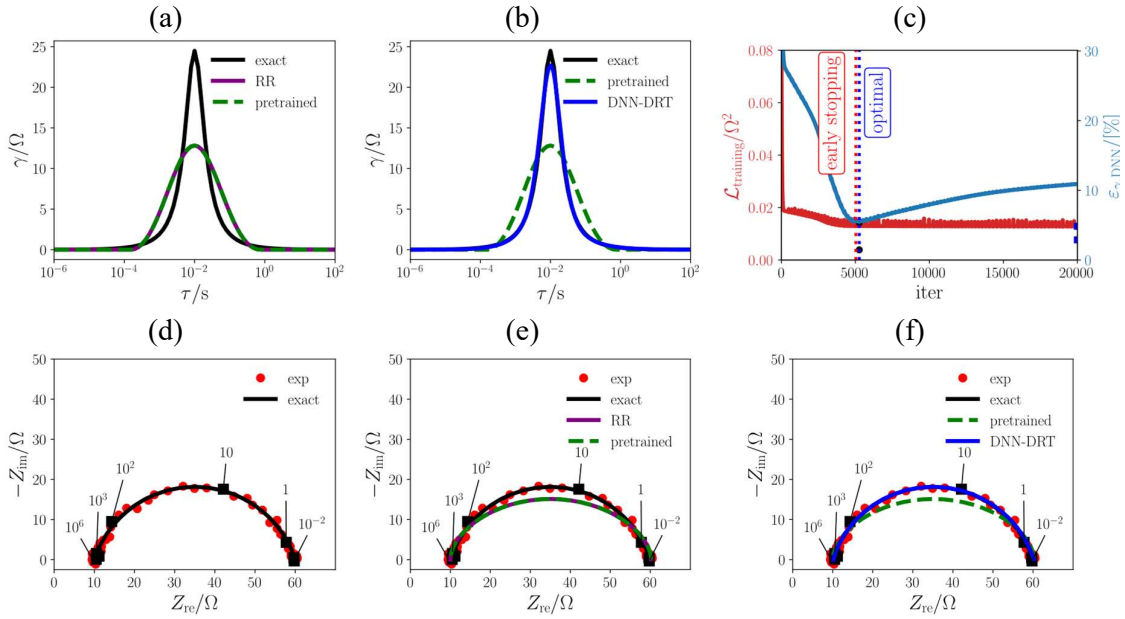

Figure 2: (a) RR and (b) trained DNN DRTs with the exact and pretrained DNN DRTs for the single ZARC model ($N = 10$ M); (c) loss function $\mathcal{L}_{\mathrm{training}}(\boldsymbol{\theta})$ and DRT error $\varepsilon_{\gamma,\mathrm{DNN}}$ as functions of the iteration number; Nyquist plots of the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

Figure 3 (a) displays the absolute value of the truncation error on the real and imaginary parts of the impedances, $|\varepsilon_{\mathrm{trunc,re}}(f)|$ and $|\varepsilon_{\mathrm{trunc,im}}(f)|$, respectively, as functions of the frequency.

$|\varepsilon_{\text{trunc,re}}(f)|$ was virtually frequency-independent up to $f = 10^5$ Hz and experienced a slight decrease after that. On the other hand, $|\varepsilon_{\text{trunc,im}}(f)|$ decreased until a minimum was reached at $f \approx 10^2$ Hz before it increased again up to $f \approx 10^5$ Hz. We also noted that $|\varepsilon_{\text{trunc,im}}(f)| < |\varepsilon_{\text{trunc,re}}(f)|$ for all frequencies. As for $|\varepsilon_{\text{integ,re}}(f)|$ and $|\varepsilon_{\text{integ,im}}(f)|$, we observed that they both reached a maximum at around $f \approx 10^{-1}$ Hz before decreasing afterwards (see Figure 3 (b)).

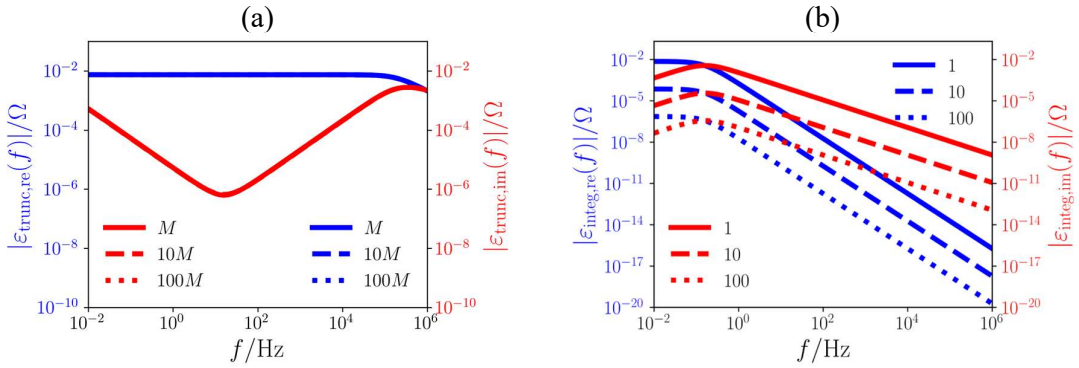

Figure 3: Absolute value of the (a) truncation and (b) integration errors on the real and imaginary parts of the single ZARC impedance as functions of the frequency for $N = M$, 10 M, and 100 M.

3.1.1.2 Influence of the Grid Density on the Truncation and Integration Errors

We set $M = 81$ and carried out simulations for $N = M$, 5 M, 10 M, …, 500 M. For all values of N, the DNN recovery of the DRT and impedance were greatly improved compared to RR (see Figure S2 and the boxplots in Figure S3). This was validated with the DRT and impedance errors ($\varepsilon_{\gamma,\text{RR}} = 42.25\%$, $\varepsilon_{\gamma,\text{DNN}} \approx 5.51\%$, $\bar{\varepsilon}_{\text{Z,RR}} = 5.69\%$, and $\bar{\varepsilon}_{\text{Z,DNN}} \approx 3.41\%$, see Table S8). Graphically, we observed for $N = M$, 10 M, and 100 M that $|\varepsilon_{\text{trunc,re}}| > |\varepsilon_{\text{trunc,im}}|$, and both errors were independent of N (Figure 3 (a)), consistent with the literature [39]. Moreover, $|\varepsilon_{\text{integ,re}}|$ and $|\varepsilon_{\text{integ,im}}|$ decreased when N increased (Figure 3(b)), consistent with intuition. Similar to Section 3.1.1.1, we validated these results with ten synthetic spectra (see Table 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 followed the same procedure as in Section 3.1.1.1 with $\sigma_n^{\exp} = 0.1$ Ω, 2 Ω, and 5 Ω. When $\sigma_n^{\exp}$ increased from 0.1 Ω to 5 Ω, we observed that $\varepsilon_{\gamma,\mathrm{RR}}$ did not vary ($\varepsilon_{\gamma,\mathrm{RR}} \approx 42.28\%$, Table S10), but $\varepsilon_{\gamma,\mathrm{DNN}}$ increased ($\varepsilon_{\gamma,\mathrm{DNN}} \approx 4.39\%$, 5.50%, 17.46%, and 26.13% for $\sigma_n^{\exp}$= 0.1, 0.5 Ω, 2 Ω, and 5 Ω, respectively). Additionally, $\mathcal{L}_{\mathrm{training}}(\boldsymbol{\theta})$ and $\varepsilon_{\gamma,\mathrm{DNN}}$ reached their minima in fewer iterations, but the minimum of $\mathcal{L}_{\mathrm{training}}(\boldsymbol{\theta})$ increased (Figure S4). This led to a decrease in the DNN accuracy of the impedance recovery ($\bar{\varepsilon}_{\mathrm{Z,RR}} = 4.01\%$, 5.69%, 14.16%, and 34.26%, and $\bar{\varepsilon}_{\mathrm{Z,DNN}} \approx 0.72\%$, 3.40%, 13.17%, and 33.98% for $\sigma_n^{\exp}$= 0.1, 0.5 Ω, 2 Ω, and 5 Ω, respectively, see Table S11). We stress that $\varepsilon_{\gamma,\mathrm{RR}}$ $\varepsilon_{\gamma,\mathrm{DNN}}$, $\bar{\varepsilon}_{\mathrm{Z,RR}}$, and $\bar{\varepsilon}_{\mathrm{Z,DNN}}$ were independent of N (see Tables S10 and S11). Lastly, we found similar averaged values for these four metrics using ten synthetic spectra for $N = M$ and each $\sigma_n^{\exp}$ (see Tables S12 and S13).

3.1.1.4 Influence of λ_{RR} on the DNN Training

RR relies on a penalty coefficient λ_{RR} (see (6)), which can significantly affect the DRT and impedance recoveries [37, 47, 52]. To investigate this phenomenon, we set $N = 10\,M$ and ran simulations for λ_{RR} =10^{-6}, 10^{-5}, …, 10^{-1}. For small values of λ_{RR} (*i.e.*, $\lambda_{\mathrm{RR}} = 10^{-6}$, 10^{-5}, and 10^{-4}), we observed oscillations at the bottom of the DRT peak (Figure S5), while large values of λ_{RR} (*i.e.*, $\lambda_{\mathrm{RR}} = 10^{-3}$, 10^{-2}, and 10^{-1}) reduced and enlarged the DRT peak (Figure S6). In other words, low and large values of λ_{RR} led to underfitting and overfitting, respectively, consistent with the literature [47, 72]. Then, we found that the trained DNN was able to match the exact DRT (Figures S5 and S6), *i.e.*, λ_{RR} did not affect the overall DNN performance. Consistently, the errors for RR $\varepsilon_{\gamma,\mathrm{RR}}$ and $\bar{\varepsilon}_{\mathrm{Z,RR}}$ varied more significantly than the corresponding DNN errors $\varepsilon_{\gamma,\mathrm{DNN}}$ and $\bar{\varepsilon}_{\mathrm{Z,DNN}}$ when λ_{RR} increased (Table S14). We confirmed these results with ten synthetic spectra (Table S15), *i.e.*, λ_{RR} only affected RR and not the DNN accuracy.

3.1.1.5 Comparison with the Deep-DRT Model

Next, we benchmarked the DNN-DRT model against the previously developed deep-DRT model for the parameters given in Section 3.1.1.1 [51]. We found that $\varepsilon_{\gamma,\mathrm{DNN}}$ and $\bar{\varepsilon}_{\mathrm{Z,DNN}}$ were lower for the DNN-DRT model ($\varepsilon_{\gamma,\mathrm{DNN}} = 7.65\%$ and $\bar{\varepsilon}_{\mathrm{Z,DNN}} = 6.67\%$ versus $\varepsilon_{\gamma,\mathrm{DNN}} = 8.06\%$ and $\bar{\varepsilon}_{\mathrm{Z,DNN}} = 8.38\%$ for the deep-DRT model), see Table S16. In other words, the new DNN architecture proposed herein enabled us to improve the DRT and impedance recoveries, which we confirmed with ten synthetic spectra (Tables S16 and S17).

3.1.2 2×ZARC Model

3.1.2.1 Overlapping, Separated, and Distant 2×ZARC Models

In this section, we assessed the ability of the DNN-DRT model to distinguish processes characterized by close timescales. First, we studied the overlapping 2×ZARC ($\tau_1 = 10^{-1}$ s, $\tau_2 = 10^{-2}$ s), separated 2×ZARC ($\tau_1 = 10^{-1}$ s, $\tau_2 = 10^{-4}$ s) and distant 2×ZARC ($\tau_1 = 10^{0}$ s, $\tau_2 = 10^{-4}$ s) models. All parameters are reported in Table S18. The left panel of Figure 4 shows for $N = 10$ M and each 2×ZARC model the exact, pretrained DNN, and trained DNN DRTs. The experimental, exact, pretrained DNN, and trained DNN impedances for each 2×ZARC model are displayed on the right panel of Figure 4. We found that the trained DNN DRT was much closer to the exact DRT than the pretrained DNN DRT (Figure 4). This was validated with the DNN error for RR ($\varepsilon_{\gamma,\mathrm{RR}} = 24.33\%$, 41.95%, and 44.49% for the overlapping, separated, and distant 2×ZARC models, respectively, see Table S4) and for DNN training ($\varepsilon_{\gamma,\mathrm{DNN}} \approx 6.60\%$, 6.75%, and 13.59%). Moreover, the trained DNN slightly improved the impedance recovery, as shown in Figure 4 and by the impedance errors for RR ($\bar{\varepsilon}_{\mathrm{Z,RR}} \approx 3.28\%$, 5.48%, and 5.34%, see Table S5) and for DNN training ($\bar{\varepsilon}_{\mathrm{Z,DNN}} \approx 3.22\%$, 2.12%, and 2.12%). We stress that $\varepsilon_{\gamma,\mathrm{RR}}$ $\varepsilon_{\gamma,\mathrm{DNN}}$, $\bar{\varepsilon}_{\mathrm{Z,RR}}$, and $\bar{\varepsilon}_{\mathrm{Z,DNN}}$ were independent of N for all models (see Tables

S4 and S5, and Figure S7). Additionally, we used ten synthetic spectra to confirm these conclusions (Tables S6 and S7, and the boxplots on Figure S7).

Second, we focused on the separated 2×ZARC model with peaks of different heights (see the parameters in Table S18). The DRTs and Nyquist plots are presented in Figure 5, and the errors in Table S4. We found the same result as for the 2×ZARC model with peaks of the same height. More specifically, the DNN training improved the DRT and impedance recoveries compared to RR, and the errors were all independent of $N = M$, 10 M and 100 M (Tables S4 and S5). These conclusions were validated for $N = M$ by generating ten synthetic spectra (Tables S6 and S7).

Third, we compared the DNN-DRT model against the DRTtools for $N = M$ [37] using the separated $2 \times$ ZARC model with peaks of different heights. The DNN-DRT model outperformed the DRTtools for the DRT recovery (see $\varepsilon_{\gamma,\mathrm{DNN}}$ in Table S19) and led to a similar impedance recovery ($\bar{\varepsilon}_{\mathrm{Z,DNN}}$ in Table S20), which we also noted graphically (Figure S8).

Furthermore, we compared the DNN-DRT and deep-DRT models for the overlapping and separated 2×ZARC models (the parameters are described in [51]). For these two 2×ZARC models, the DNN-DRT model only improved the impedance recovery (Table S16), which was confirmed with ten synthetic spectra (Tables S16 and S17).

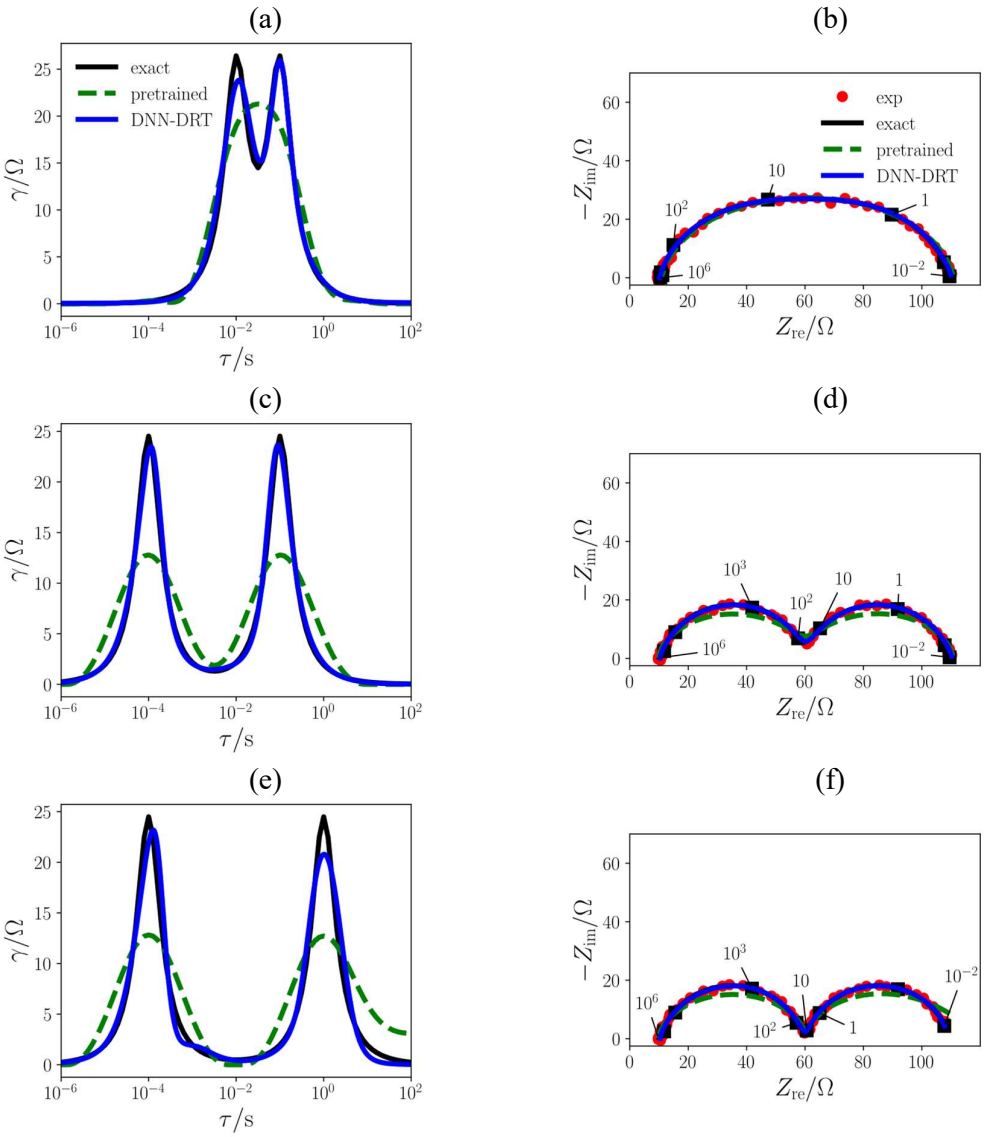

Figure 4: Exact, pretrained DNN, and trained DNN DRTs for the (a) overlapping, (c) separated, and (e) distant 2×ZARC models (N =10 M); corresponding Nyquist plots (b), (d), and (f) with the experimental impedances.

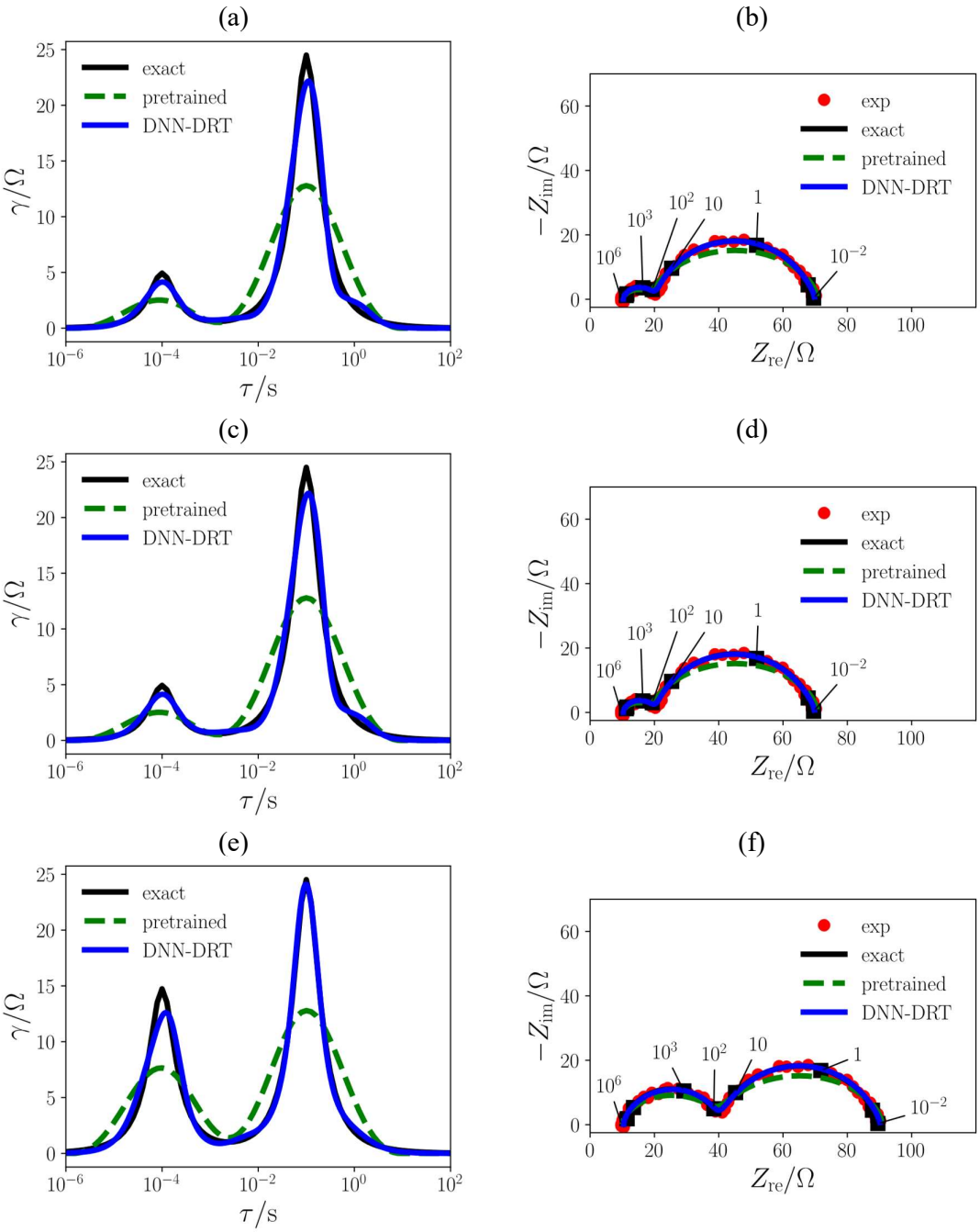

Figure 5: Exact, pretrained DNN, and trained DNN DRTs for the separated 2×ZARC model with $R_{\mathrm{ct},2}$ = (a) 10 Ω, (c) 30 Ω, and (e) 50 Ω (N =10 M); corresponding Nyquist plots (b), (d), and (f) with the experimental impedances.

3.1.2.2 Truncation Errors

Unlike for the single ZARC model (Section 3.1.1.2), the truncation error, $\bar{\varepsilon}_{\mathrm{trunc}}$, for the separated 2×ZARC model with two peaks of the same height (Section 3.1.2.1) remained small but not negligible compared to the averaged DNN impedance error, $\bar{\varepsilon}_{\mathrm{Z,DNN}}$ ($\bar{\varepsilon}_{\mathrm{trunc}} \approx 0.92\%$, $\varepsilon_{\gamma,\mathrm{RR}} = 41.95\%$, $\varepsilon_{\gamma,\mathrm{DNN}} \approx 6.75\%$, $\bar{\varepsilon}_{\mathrm{Z,RR}} = 5.48\%$, $\bar{\varepsilon}_{\mathrm{Z,DNN}} \approx 2.12\%$ for $N = 1$, 10, and 100 M, see Tables S4 and S5). In this section, we studied the impact of the integration bounds τ_1 and τ_N on $\bar{\varepsilon}_{\mathrm{trunc}}$, and, in particular, we set $\log \tau_1$ =-8 and $\log \tau_N$ =4. We found that $\bar{\varepsilon}_{\mathrm{trunc}}$ was remarkably reduced ($\bar{\varepsilon}_{\mathrm{trunc}}$ =2.57×10^{-2}%, see Table S20). In other words, extending the range of timescales led to a significant reduction on $\bar{\varepsilon}_{\mathrm{trunc}}$. Interestingly, the errors $\varepsilon_{\gamma,\mathrm{RR}}$ and $\bar{\varepsilon}_{Z,\mathrm{RR}}$ for RR increased while the errors for DNN training reduced for all values of N ($\varepsilon_{\gamma,\mathrm{RR}} =$ 48.90%, $\varepsilon_{\gamma,\mathrm{DNN}} = 5.92\%$, $\bar{\varepsilon}_{Z,\mathrm{RR}} = 7.84\%$, $\bar{\varepsilon}_{Z,\mathrm{DNN}} \approx 2.09\%$ for $N = 1$, 10, and 100 M).

3.1.3 Discontinuous DRTs

Next, we tested how well the DNN-DRT model could recover discontinuous DRTs by studying the fractal, PWC, and Gerischer models (the parameters are given in Tables S21, S22, and S23, respectively). For all three models and $N = 1$, 10, and 100 M, the peak was approximately identified during RR, as shown in Figures 6, S9, and S10. This was confirmed by the large values of $\varepsilon_{\gamma,\mathrm{RR}}$ ($\varepsilon_{\gamma,\mathrm{RR}} = 77.15\%$, 29.51%, and 67.95% for the fractal, PWC, and Gerischer models, respectively, see Table S4 for all N). Training the DNN enabled us to increase the accuracy, as evidenced by the lower values $\varepsilon_{\gamma,\mathrm{DNN}}$ compared to $\varepsilon_{\gamma,\mathrm{RR}}$ ($\varepsilon_{\gamma,\mathrm{DNN}} \leq 75.89\%$, 16.99%, and 66.07%, see Table S4). Furthermore, when N increased from M to 10 M, $\varepsilon_{\gamma,\mathrm{DNN}}$ decreased for the fractal and Gerischer models but increased for the PWC model; then, $\varepsilon_{\gamma,\mathrm{DNN}}$ increased for all three models when N varied from 10 M to 100 M (Table S4). Similarly, the impedance recovery was slightly improved by DNN training compared to RR, see Figures 6, S9, and S10, and the values of the impedance errors ($\bar{\varepsilon}_{\mathrm{Z,RR}} = 6.94\%$, 5.37%, and 6.00%, versus $\bar{\varepsilon}_{\mathrm{Z,DNN}} \leq$6.91%, 4.17%, and 4.16%, see Table S5). Interestingly, $\bar{\varepsilon}_{\mathrm{Z,DNN}}$ decreased for the

fractal model and did not vary for the PWC and Gerischer models, see Table S5. We validated these results by drawing ten synthetic EIS spectra (Tables S6 and S7, and the boxplots in Figure S11).

Next, we compared the DNN-DRT model against the DRTtools for $N = M$. Both led to similar results for the DRT recovery of the fractal model ($\varepsilon_{\gamma,\mathrm{DNN}} = 73.94\%$ and 73.69% for the DNN-DRT and DRTtools, respectively, see Table S19), and for the impedance recovery of the PWC ($\bar{\varepsilon}_{Z,\mathrm{DNN}} = 4.17\%$ and 4.16%) and Gerischer ($\bar{\varepsilon}_{Z,\mathrm{DNN}} = 4.14\%$ and 4.03%) models (see Figure S8). We also noted that the DRTtools outperformed the DNN-DRT model for the impedance recovery of the fractal model ($\bar{\varepsilon}_{Z,\mathrm{DNN}} = 6.91\%$ and 4.46%). However, the DNN-DRT model excelled for the DRT recovery of the PWC ($\varepsilon_{\gamma,\mathrm{DNN}} = 10.16\%$ and 23.03%) and Gerischer ($\varepsilon_{\gamma,\mathrm{DNN}} = 59.53$ % and 61.02%) models.

In addition, we compared the DNN-DRT and deep-DRT models for $N = M$ (the parameters are reported in Table S16) for the fractal and PWC models. The DNN-DRT model outperformed the deep-DRT model for the DRT recovery of the fractal ($\varepsilon_{\gamma,\mathrm{DNN}} = 73.30\%$ and 89.47% for the DNN-DRT and deep DRT models, respectively, see Table S16) and PWC ($\varepsilon_{\gamma,\mathrm{DNN}} = 7.50\%$ and 10.45%) models. We also found that the DNN-DRT and deep-DRT models performed similarly for the impedance recovery ($\bar{\varepsilon}_{Z,\mathrm{DNN}} = 6.91\%$ and 6.00% for the fractal model, $\bar{\varepsilon}_{Z,\mathrm{DNN}} = 7.30\%$ and 7.83% for the PWC model). The comparisons between the DNN-DRT and deep-DRT models were validated with ten synthetic spectra for each impedance model investigated (Table S17).

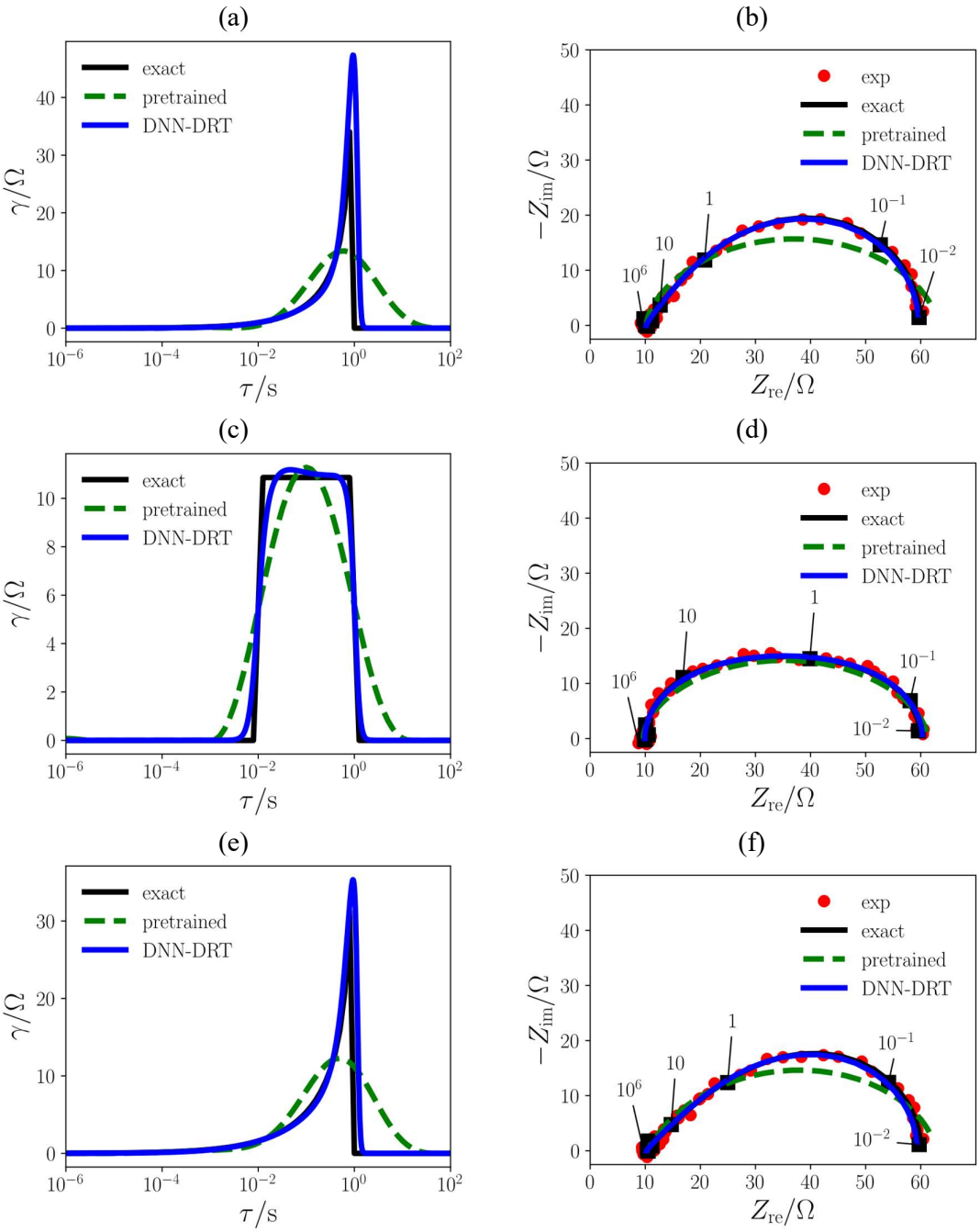

Figure 6: Exact, pretrained DNN, and trained DNN DRTs for the (a) fractal, (c) PWC, and (e) Gerischer models ($N = 10\ M$); corresponding Nyquist plots (b), (d), and (f) with the experimental impedances.

3.1.4 DRTs with Inductive Features

3.1.4.1 Preliminary Analysis

Next, we focused on the ability of the DNN-DRT model to handle DRTs with negative peaks, which we described with the hook model (the parameters are reported in Table S24 [61]). After RR, we observed oscillations at the bottom of each peak, and both peaks were underestimated (Figure 7 (a)). This explains the high value of the DRT error for RR ($\varepsilon_{\gamma,\mathrm{RR}} = 41.30\%$, see Table S4). Then, the DRT recovery was greatly improved after DNN training because of the two additional penalty terms $P_{\partial}(\boldsymbol{\theta})$ and $P_{\mathrm{cross}}(\boldsymbol{\theta})$ in $\mathcal{L}_{\mathrm{training}}(\boldsymbol{\theta})$ (see (9)), which led to $\varepsilon_{\gamma,\mathrm{DNN}} \leq$ 34.24%. The specific role of each penalty term will be further investigated in the next section. Regarding the impedance, we noted that RR was very accurate, as can be understood graphically (Figure Figure 7 (b)) and with the average impedance error ($\bar{\varepsilon}_{\mathrm{Z,RR}} = 2.27\%$ and $\bar{\varepsilon}_{\mathrm{Z,DNN}} \leq 3.00\%$, see Table S5). These findings did not vary with the number of collocation points N (see the values of each error for $N = M$, 10 M, and 100 M in Tables S4 and S5, and in Figure S12), and we validated them with ten synthetic spectra (Tables S6 and S7, and the boxplots in Figure S11).

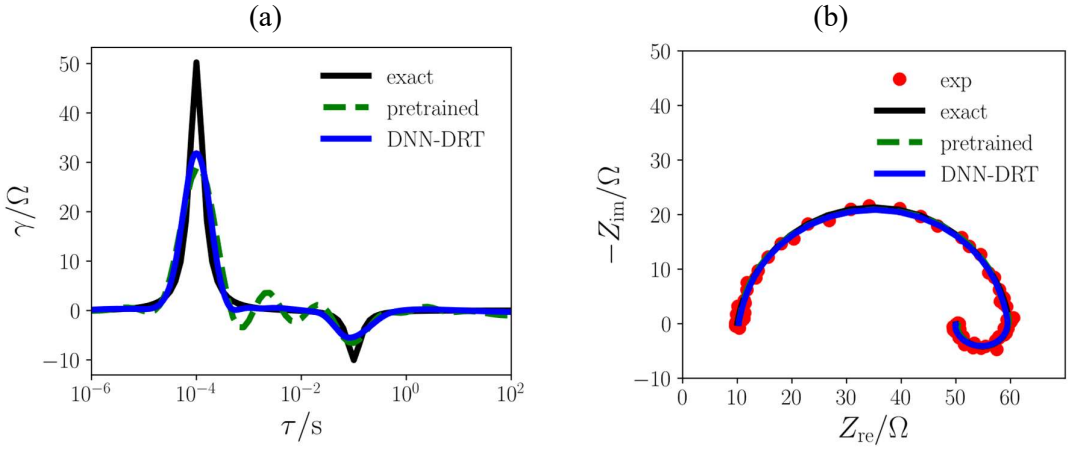

Figure 7: (a) Exact, pretrained DNN, and trained DNN DRTs for the hook model (N =10 M) and (b) corresponding Nyquist plot with the experimental impedance.

3.1.4.2 Effect of the Penalty Functions on the Recovery of Inductive DRTs

In this section, we studied the effect of the penalties P_∂ and P_{cross} added in $\mathcal{L}_{\text{training}}(\boldsymbol{\theta})$ to handle DRTs with negative peaks. More precisely, we investigated for $N = 10\ M$ P_∂ and each component of P_{cross} in (11) to assess how well these functions could suppress oscillations and minimize the errors $\varepsilon_{\gamma,\text{DNN}}$ and $\varepsilon_{Z,\text{DNN}}$. Interestingly, P_∂ alone was not sufficient to suppress the oscillations created during RR, and we found in this case that $\varepsilon_{\gamma,\text{DNN}}$ and $\bar{\varepsilon}_{\text{Z,DNN}}$ were close to the case without penalties (Table S25). Moreover, the minimum of $\varepsilon_{\gamma,\text{DNN}}$ was achieved using the second term in P_{cross}, while the finest impedance recovery (*i.e.*, lowest $\bar{\varepsilon}_{\text{Z,DNN}}$) was obtained by combining P_∂ and only the first term in P_{cross}.

3.1.5 Investigation of the Influence of the DNN Inputs

To further validate our error analysis, we tested the trained DNN under the following two scenarios. First, we kept the initial $\log \boldsymbol{\tau}$ and chose 81 log-spaced frequencies between $10^{-1.9}$ Hz and $10^{5.9}$ Hz for the set of frequencies $\boldsymbol{f}$ to obtain the experimental impedance (Figure 1). Second, given the initial $\boldsymbol{f}$, we chose 79 equispaced points between -5.4 and 1.8 with the same original bounds -6 and 2 for $\log \boldsymbol{\tau}$ (81 log-timescales in total) to deconvolve the DRT. We then investigated the single ZARC, PWC, and hook models with $N = M$, $10\ M$, and $100\ M$. Compared to the results for the single ZARC model in Section 3.1.1.1 ($\varepsilon_{\gamma,\text{DNN}} \approx 5.50\%$ and $\bar{\varepsilon}_{Z,\text{DNN}} \approx 3.40\%$ in Tables S4 and S5 for $N = M$, $10\ M$, and $100\ M$), the DRT and impedance errors did not vary for either scenario and all values of N ($\varepsilon_{\gamma,\text{DNN}} \approx 5.66\%$ and $\bar{\varepsilon}_{Z,\text{DNN}} \approx 3.39\%$, Table S26). Regarding the PWC model, we noticed a large increase in $\varepsilon_{\gamma,\text{DNN}}$ when we changed $\log \boldsymbol{\tau}$ for $N = M$ (from 10.16% to 17.81%, Table S26). Otherwise, we did not observe any change in either $\varepsilon_{\gamma,\text{DNN}}$ or $\bar{\varepsilon}_{\text{Z,DNN}}$. As for the hook model, only the second scenario led to an increase in $\bar{\varepsilon}_{\text{Z,DNN}}$ for all N (Table S26).

3.1.6 Discussion

To summarize the results obtained with synthetic EIS data, we first noted that the DNN training improved the DRT and impedance recoveries compared to RR for all cases except the hook model. Using the single ZARC model, we showed that i) the DRT and impedance errors were negligibly affected by the number of collocation points; ii) the DRT and impedance recoveries became less precise as the level of experimental noise increased; and iii) the regularization parameter for RR did not affect the DNN performance. We also showed that this new DNN architecture can recover various impedances, including those with negative peaks. Furthermore, the DNN-DRT model performed as well or even better than the deep-DRT model and DRTtools in the various cases investigated. Lastly, we confirmed the conclusion drawn for each synthetic case using ten synthetic experiments.

3.2 Deconvolution of Experimental Data

We evaluated the DRT-DNN model using EIS data collected from real electrochemical systems. Specifically, we investigated EIS spectra obtained from a lithium-ion battery, a solid oxide fuel cell, and a proton exchange membrane fuel cell.

3.2.1 Lithium-ion Battery

We started with a symmetric cell built with lithium-ion electrodes and a composite polymer electrolyte [59]. The impedance was modeled with a $3 \times$ZARC whose parameters were estimated by RR (see Table S27 for the values of these parameters). Given $N = 10\ M$, Figure 9 presents the recovered DRT and impedance, while Figure S13 shows the plots of the impedance modulus and argument. Regarding the DRT, the peaks at $\tau_1 = 1$ s and $\tau_2 = 1.91 \times 10^{-4}$ s were well identified. However, the peak at $\tau_3 = 1.05 \times 10^{-7}$ s was not recovered because the corresponding dispersion factor ϕ_3 was equal to 1 (Table S27). This explains the high DRT error for RR ($\varepsilon_{\gamma,\mathrm{RR}} = 44.17\%$ for all N, see Table S28), which was greatly reduced

after DNN training ($\varepsilon_{\gamma,\mathrm{DNN}} \approx 26.17\%$). The same conclusion can be made for the impedance recovery ($\bar{\varepsilon}_{\mathrm{Z,RR}} = 9.30\%$ and $\bar{\varepsilon}_{\mathrm{Z,DNN}} \approx 6.09\%$, see Table S29).

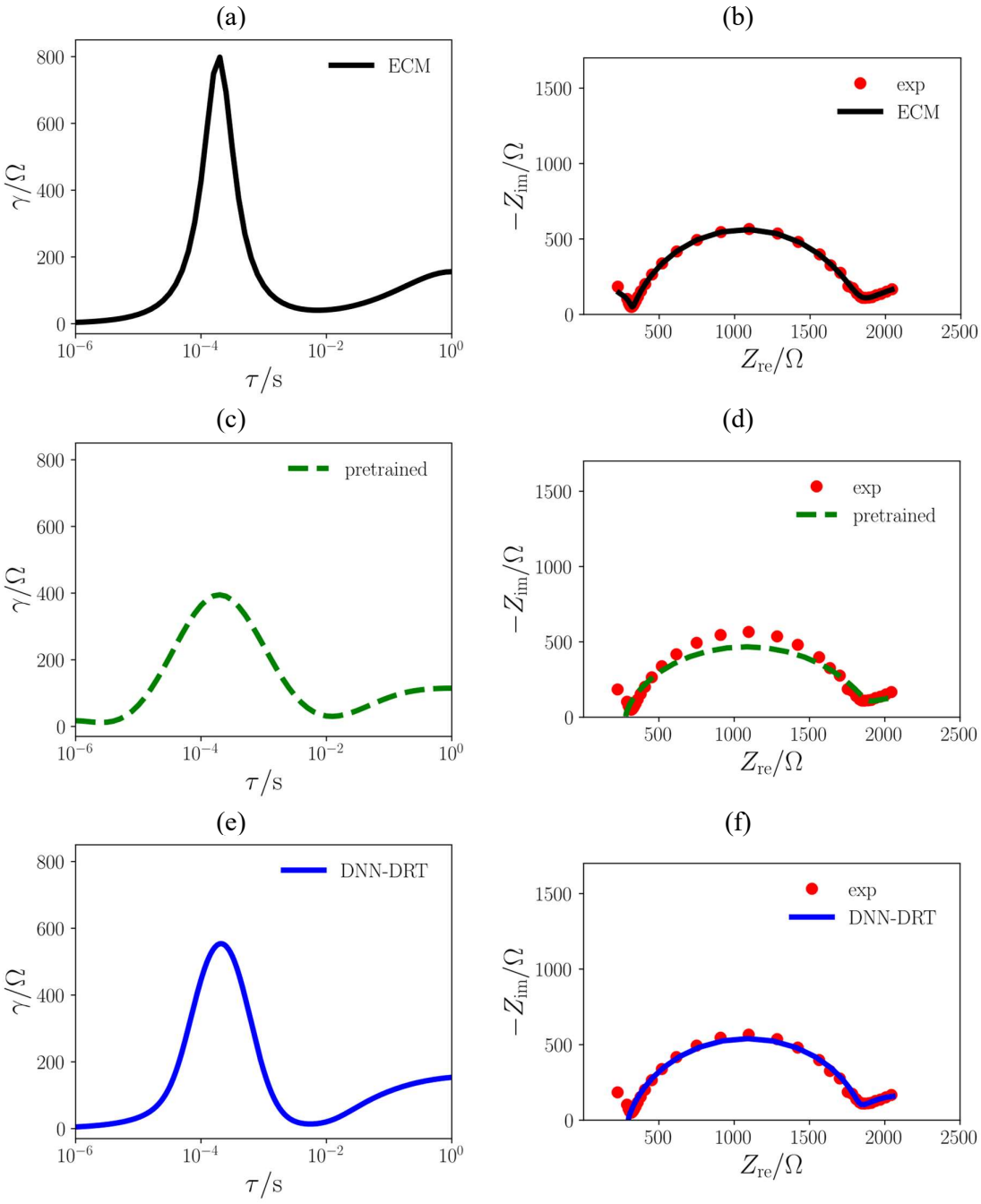

Figure 8: (a) Fitted-ECM, (c) pretrained DNN, and (e) trained DNN DRTs for a lithium-ion symmetric cell ($N = 10\ M$); corresponding Nyquist plots (b), (d), and (f).

3.2.2 Fuel Cells

3.2.2.1 Solid Oxide Fuel Cell

Secondly, we analyzed the EIS spectrum from a symmetric solid oxide fuel cell with $(Ba_{0.95}La_{0.05})_{0.95}FeO_{3-\delta}$ for the cathode material and samarium-doped ceria for the electrolyte [51, 60]. Following [73], we modeled the experimental impedance with a 2×ZARC model whose parameters are given in Table S30. After RR, the position of the DRT peak was well identified but its height was slightly underestimated (Figures Figure 9 and S13 for $N = 10\,M$), hence the large DRT error for RR $\varepsilon_{\gamma,\mathrm{RR}} = 20.72\%$ (Table S28). Comparatively, the impedance error was negligible ($\bar{\varepsilon}_{\mathrm{Z,RR}} = 0.91\%$, Table S29). Then, the trained DNN remarkably reduced both errors ($\varepsilon_{\gamma,\mathrm{DNN}} = 13.25\%$ and $\bar{\varepsilon}_{\mathrm{Z,DNN}} = 0.18\%$). Notably, all errors did not depend on $N = M$, $10\,M$, and $100\,M$ (Tables S28 and S29).

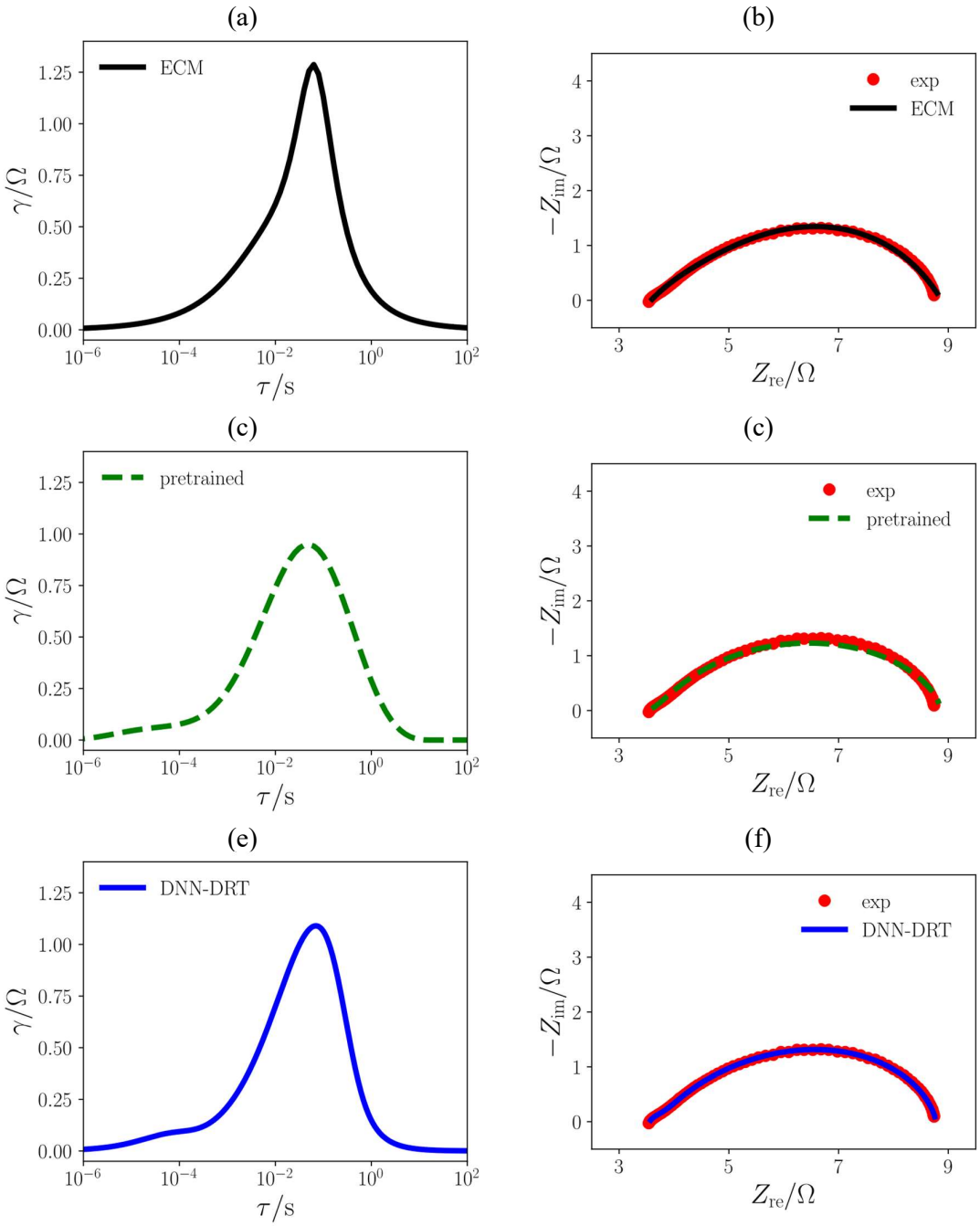

Figure 9: (a) Fitted-ECM, (c) pretrained DNN, and (e) trained DNN DRTs for the solid oxide fuel cell ($N = 10\ M$); corresponding Nyquist plots (b), (d), and (f).

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 Nafion electrolyte membrane with an

inductive component [61]. 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 concluded that the trained DNN closely deconvolved the DRT and fitted the EIS spectrum, see Figures 10 and S13 for $N = 10\ M$ and the errors ($\varepsilon_{\gamma,\mathrm{RR}} = 29.65\%$, $\varepsilon_{\gamma,\mathrm{DNN}} \leq 27.35\%$, $\bar{\varepsilon}_{\mathrm{Z,RR}} = 1.95\%$, and $\bar{\varepsilon}_{\mathrm{Z,DNN}} \leq 3.19\%$ in Tables S28 and S29).

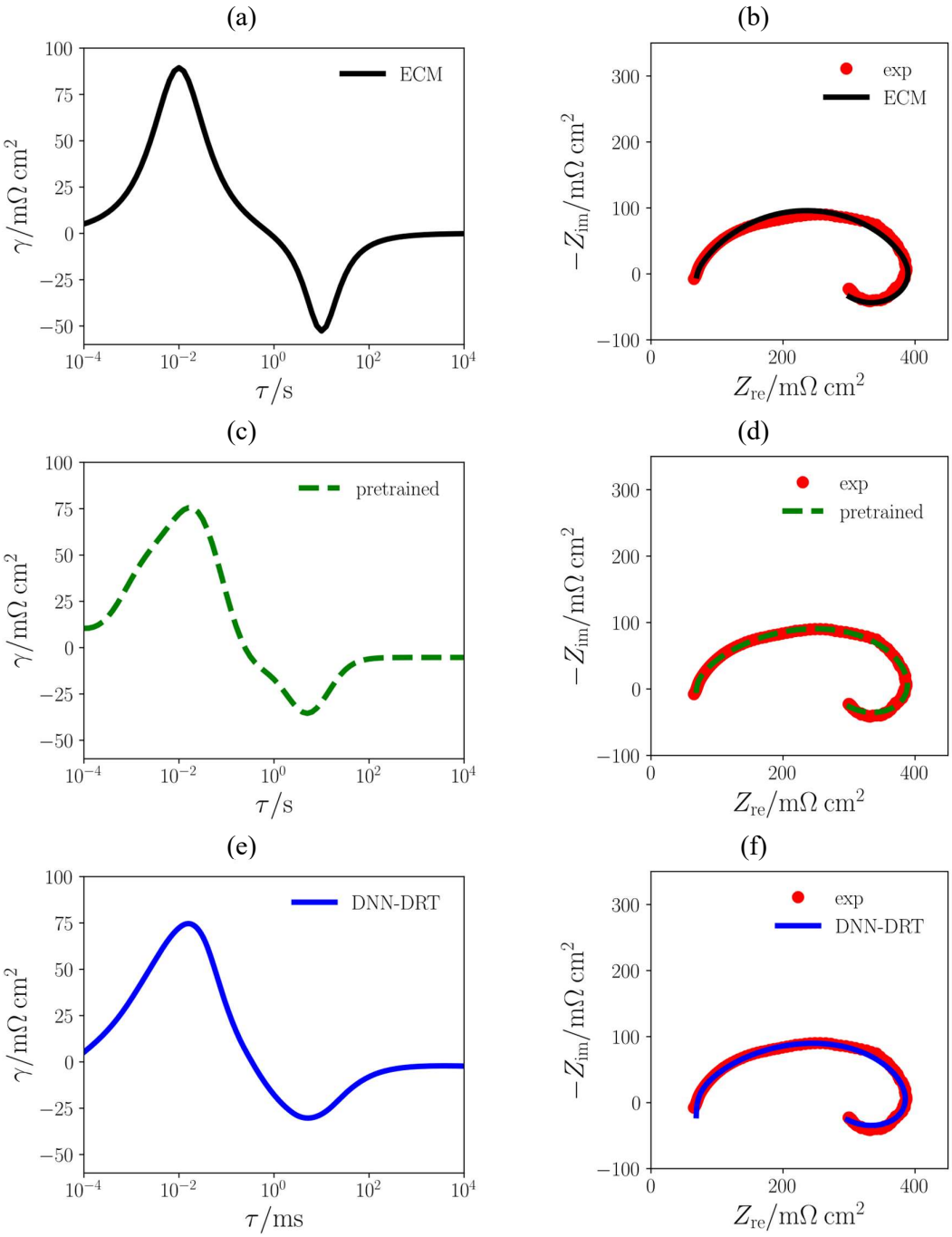

Figure 10: (a) Fitted-ECM, (c) pretrained DNN, and (e) trained DNN DRTs for the proton exchange membrane fuel cell ($N = 10\ M$); corresponding Nyquist plots (b), (d), and (f).

4 Conclusions

For the purpose of analyzing EIS data, the DRT has been successfully deconvolved using many approaches, including the use of neural networks. Leveraging a new architecture and a

pretraining step, this work successfully addresses three main limitations of DNNs, namely the long training time, the fixed grid density to compute the DRT integral, and the deconvolution of inductive DRTs. Using synthetic and real EIS data, we show how the DNN-DRT model can solve these challenges, and in particular a thorough error analysis is proposed to evaluate the DRT and impedance recoveries. We also demonstrate that this new model performs at least as well as the DRTtools and the deep-DRT model. Overall, we envision that this work will foster the implementation of new DNN architectures to improve the analysis of EIS data for energy systems.

Author Credit Statement

Emanuele Quattrocchi: Extension, Development, and Validation of software, Methodology, Theory, Data Curation, Formal Analysis, Experimental Analysis, Writing – Original Draft, Writing –, extensive Review & Editing. Baptiste Py: Methodology, Theory, Investigation, Writing, extensive Review & Editing. Adeleke Maradesa: Investigation, Writing, Review & Editing. Quentin Meyer & Chuan Zhao: Experimental Investigation, Review & Editing. Francesco Ciucci: Software Creation, Conceptualization, Methodology, Theory development, Investigation, Resources, Review & Editing, Funding acquisition, Project administration, Supervision.

Declaration of Competing Interest

None.

Acknowledgement

The authors gratefully acknowledge the financial support from the Research Grants Council of Hong Kong (RGC Ref. No. 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). C. Zhao acknowledges the Australian Research Council for financial support (FT170100224, LP200100255, IC200100023). Quattrocchi and A. Maradesa kindly thank the Hong Kong PhD. Fellowship Scheme for its financial support.

List of Symbols

$\boldsymbol{\gamma}$	Vector of discretized DRTs
$\boldsymbol{\varepsilon}$	Vector of experimental errors
$\boldsymbol{\theta}$	Vector of DNN parameters (biases and weights)
$\boldsymbol{\tau}$	Vector of timescales
γ	DRT
$\varepsilon_{\text{integ}}$	Integration error
ε_{γ}	Normalized DRT error
$\bar{\varepsilon}_{\text{Z}}$	Average impedance error with respect to the frequency
$\varepsilon_{\text{trunc}}$	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

$\boldsymbol{f}$	Vector of frequencies
$\mathbf{I}$	Identity matrix
$\boldsymbol{Z}$	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_{∂}	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 RR, DNN training, and experimental, respectively.

List of 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

References

[1] I. Babaeiyazdi, A. Rezaei-Zare, S. Shokrzadeh, State of charge prediction of EV Li-ion batteries using EIS: a machine learning approach, Energy, 223 (2021) 120116.

[2] M. Koseoglou, E. Tsioumas, D. Papagiannis, N. Jabbour, C. Mademlis, A novel on-board electrochemical impedance spectroscopy system for real-time battery impedance estimation, IEEE Transactions on Power Electronics, 36 (2021) 10776-10787.

[3] J.P. Schmidt, P. Berg, M. Schönleber, A. Weber, E. Ivers-Tiffée, The distribution of relaxation times as basis for generalized time-domain models for Li-ion batteries, Journal of Power Sources, 221 (2013) 70-77.

[4] L. Wildfeuer, P. Gieler, A. Karger, Combining the distribution of relaxation times from EIS and time-domain data for parameterizing equivalent circuit models of lithium-ion batteries, Batteries, 7 (2021) 52.

[5] M.A. Danzer, Generalized Distribution of Relaxation Times Analysis for the Characterization of Impedance Spectra, Batteries, 5 (2019) 53.

[6] B. Kim, I.S. Chang, R.M. Dinsdale, A.J. Guwy, Accurate measurement of internal resistance in microbial fuel cells by improved scanning electrochemical impedance spectroscopy, Electrochimica Acta, 366 (2021) 137388.

[7] A. Iranzo, S.J. Navas, F. Rosa, M.R. Berber, Determination of time constants of diffusion and electrochemical processes in polymer electrolyte membrane fuel cells, Energy, 221 (2021) 119833.

[8] A. Baldinelli, A. Staffolani, G. Bidini, L. Barelli, F. Nobili, An extensive model for renewable energy electrochemical storage with solid oxide cells based on a comprehensive analysis of impedance deconvolution, Journal of Energy Storage, 33 (2021) 102052.

[9] J. Zhang, L. Lei, H. Li, K. Sun, Z. Sun, M. Han, Experimental investigations of cell resistances to characterize the concentration polarization behavior of 10 ×10 cm^2 solid oxide fuel cells, Journal of Power Sources, 516 (2021) 230678.

[10] 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, Journal of Power Sources, 470 (2020) 228285.

[11] S.B. Davey, A.P. Cameron, K.G. Latham, S.W. Donne, Combined step potential electrochemical spectroscopy and electrochemical impedance spectroscopy analysis of the glassy carbon electrode in an aqueous electrolyte, Electrochimica Acta, 396 (2021) 139220.

[12] X.F. Sánchez-Romate, A.D. Bosque, J. Artigas-Arnaudas, B.K. Muñoz, M. Sánchez, A. Ureña, A proof of concept of a structural supercapacitor made of graphene coated woven carbon fibers: EIS study and mechanical performance, Electrochimica Acta, 370 (2021) 137746.

[13] H.D. Yoo, J.H. Jang, J.H. Ryu, Y. Park, S.M. Oh, Impedance analysis of porous carbon electrodes to predict rate capability of electric double-layer capacitors, Journal of Power Sources, 267 (2014) 411-420.

[14] L.E. Helseth, Modelling supercapacitors using a dynamic equivalent circuit with a distribution of relaxation times, Journal of Energy Storage, 25 (2019) 100912.

[15] T.M. Uehara, F.L. Migliorini, M.H.M. Facure, N.B. Palma Filho, P.B. Miranda, V. Zucolotto, D.S. Correa, Nanostructured scaffolds containing graphene oxide for nanomedicine applications, Polymers for Advanced Technologies, 33 (2021) 591-600.

[16] A. Goida, Y. Kuzin, V. Evtugyn, A. Porfireva, G. Evtugyn, T. Hianik, Electrochemical sensing of Idarubicin—DNA Interaction Using Electropolymerized Azure B and Methylene Blue Mediation, Chemosensors, 10 (2022) 33.

[17] G.L. Goh, M.F. Tay, J.M. Lee, J.S. Ho, L.N. Sim, W.Y. Yeong, T.H. Chong, Potential of printed electrodes for electrochemical impedance spectroscopy (EIS): toward membrane fouling detection, Advanced Electronic Materials, 7 (2021) 2100043.

[18] Y. Liang, F. Gao, L. Wang, S. Lin, In-situ monitoring of polyelectrolytes adsorption kinetics by electrochemical impedance spectroscopy: application in fabricating nanofiltration membranes via layer-by-layer deposition, Journal of Membrane Science, 619 (2021) 118747.

[19] S. Wang, J. Zhang, O. Gharbi, V. Vivier, M. Gao, M.E. Orazem, Electrochemical impedance spectroscopy, Nature Reviews Methods Primers, 1 (2021).

[20] A.J. Bard, L.R. Faulkner, J. Leddy, C.G. Zoski, Electrochemical methods: fundamentals and applications, Wiley New York1980.

[21] Y. Zheng, Z. Shi, D. Guo, H. Dai, X. Han, A simplification of the time-domain equivalent circuit model for lithium-ion batteries based on low-frequency electrochemical impedance spectra, Journal of Power Sources, 489 (2021) 229505.

[22] J. Fleig, Impedance spectroscopy on solids: The limits of serial equivalent circuit models, Journal of Electroceramics, 13 (2004) 637-644.

[23] M. Kunaver, M. Žic, I. Fajfar, T. Tuma, Á. Bűrmen, V. Subotić, Ž. Rojec, Synthesizing electrically equivalent circuits for use in electrochemical impedance spectroscopy through grammatical evolution, Processes, 9 (2021) 1859.

[24] J. Liu, F. Ciucci, Modeling the impedance spectra of mixed conducting thin films with exposed and embedded current collectors, Physical Chemistry Chemical Physics, 19 (2017) 26310-26321.

[25] A. Lasia, Electrochemical impedance spectroscopy and its applications, Springer2014.

[26] B.A. Boukamp, Derivation of a Distribution Function of Relaxation Times for the (fractal) Finite Length Warburg, Electrochimica Acta, 252 (2017) 154-163.

[27] H. Schichlein, A.C. Müller, M. Voigts, A. Krügel, E. Ivers-Tiffée, Deconvolution of electrochemical impedance spectra for the identification of electrode reaction mechanisms in solid oxide fuel cells, Journal of Applied Electrochemistry, 32 (2002) 875-882.

[28] B.A. Boukamp, A. Rolle, R.N. Vannier, R.K. Sharma, E. Djurado, Electrostatic spray deposited $Ca_3Co_4O_{9+\delta}$ and $Ca_3Co_4O_{9+\delta}/Ce_{0.9}Gd_{0.1}O_{1.95}$ cathodes for SOFC, Electrochimica Acta, 362 (2020) 137142.

[29] M. Hahn, S. Schindler, L.-C. Triebs, M.A. Danzer, Optimized process parameters for a reproducible distribution of relaxation times analysis of electrochemical systems, Batteries, 5 (2019) 43.

[30] F. Ciucci, C. Chen, Analysis of electrochemical impedance spectroscopy data using the distribution of relaxation times: a Bayesian and hierarchical Bayesian approach, Electrochimica Acta, 167 (2015) 439-454.

[31] B.A. Boukamp, A. Rolle, Analysis and application of distribution of relaxation times in solid state ionics, Solid State Ionics, 302 (2017) 12-18.

[32] E. Ivers-Tiffèe, A. Weber, Evaluation of electrochemical impedance spectra by the distribution of relaxation times, Journal of the Ceramic Society of Japan, 125 (2017) P4-1-P4-4.

[33] T. Reshetenko, A. Kulikovsky, Understanding the distribution of relaxation times of a low–Pt PEM fuel cell, Electrochimica Acta, 391 (2021) 138954.

[34] H. Yuan, H. Dai, P. Ming, X. Wang, X. Wei, Quantitative analysis of internal polarization dynamics for polymer electrolyte membrane fuel cell by distribution of relaxation times of impedance, Applied Energy, 303 (2021) 117640.

[35] E. Barsoukov, J.R. Macdonald, Impedance spectroscopy theory, experiment, and applications, 2nd ed., Wiley 2005.

[36] R. Franke-Lang, J. Kowal, Analysis of electrochemical impedance spectroscopy on Zinc-air batteries using the distribution of relaxation times, Batteries, 7 (2021) 56.

[37] T.H. Wan, M. Saccoccio, C. Chen, F. Ciucci, Influence of the discretization methods on the distribution of relaxation times deconvolution: implementing radial basis functions with DRTtools, Electrochimica Acta, 184 (2015) 483-499.

[38] N. Schlüter, S. Ernst, U. Schröder, Finding the optimal regularization parameter in distribution of relaxation times analysis, ChemElectroChem, 6 (2019) 6027-6037.

[39] 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 Problems, 29 (2013) 045006.

[40] B.A. Boukamp, Fourier transform distribution function of relaxation times; application and limitations, Electrochimica Acta, 154 (2015) 35-46.

[41] K. Kobayashi, T.S. Suzuki, Distribution of relaxation time analysis for non-ideal immittance spectrum: discussion and progress, Journal of the Physical Society of Japan, 87 (2018) 094002.

[42] A.K. Baral, Y. Tsur, Impedance spectroscopy of Gd-doped ceria analyzed by genetic programming (ISGP) method, Solid State Ionics, 304 (2017) 145-149.

[43] 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, International Journal of Energy Research, 45 (2021) 15948-15961.

[44] 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 Transactions on Dielectrics and Electrical Insulation, 8 (2001) 310-320.

[45] 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, Journal of Electrochemical Science and Technology, 12 (2021) 217-224.

[46] F. Dion, A. Lasia, The use of regularization methods in the deconvolution of underlying distributions in electrochemical processes, Journal of Electroanalytical Chemistry, 475 (1999) 28-37.

[47] M. Saccoccio, 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, Electrochimica Acta, 147 (2014) 470-482.

[48] T. Paul, P.W. Chi, P.M. Wu, M.K. Wu, Computation of distribution of relaxation times by Tikhonov regularization for Li ion batteries: usage of L-curve method, Sci Rep, 11 (2021) 12624.

[49] Y. Zhang, Y. Chen, M. Li, M. Yan, M. Ni, C. Xia, A high-precision approach to reconstruct distribution of relaxation times from electrochemical impedance spectroscopy, Journal of Power Sources, 308 (2016) 1-6.

[50] J. Liu, F. Ciucci, The deep-prior distribution of relaxation times, Journal of The Electrochemical Society, 167 (2020) 026506.

[51] E. Quattrocchi, 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, Electrochimica Acta, 392 (2021) 139010.

[52] 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, Electrochimica Acta, 313 (2019) 570-583.

[53] J. Huang, M. Papac, R. O'Hayre, Towards robust autonomous impedance spectroscopy analysis: a calibrated hierarchical Bayesian approach for electrochemical impedance spectroscopy (EIS) inversion, Electrochimica Acta, 367 (2021) 137493.

[54] M. Belkin, D. Hsu, S. Ma, S. Mandal, Reconciling modern machine-learning practice and the classical bias-variance trade-off, Proceedings of the National Academy of Sciences, 116 (2019) 15849-15854.

[55] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, I. Sutskever, Deep double descent: where bigger models and more data hurt, arXiv preprint arXiv:1912.02292, (2019).

[56] 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).

[57] D. Choi, H. Cho, W. Rhee, On the difficulty of DNN hyperparameter optimization using learning curve prediction, TENCON 2018-2018 IEEE Region 10 Conference, IEEE, 2018, pp. 0651-0656.

[58] D. Erhan, A. Courville, Y. Bengio, P. Vincent, Why does unsupervised pre-training help deep learning?, Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 201-208.

[59] 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, Journal of Materials Chemistry A, 7 (2019) 17995-18002.

[60] A. Belotti, Y. Wang, A. Curcio, J. Liu, E. Quattrocchi, S. Pepe, F. Ciucci, The influence of A-site deficiency on the electrochemical properties of (Ba0.95La0.05)1-xFeO3-δ as an intermediate temperature solid oxide fuel cell cathode, International Journal of Hydrogen Energy, (2021).

[61] Q. Meyer, C. Zhao, Air perturbation-induced low-frequency inductive electrochemical impedance arc in proton exchange membrane fuel cells, Journal of Power Sources, 488 (2021) 229245.

[62] D. Calvetti, E. Somersalo, Computational mathematical modeling: an integrated approach across scales, SIAM 2012.

[63] D. Klotz, Negative capacitance or inductive loop? – A general assessment of a common low frequency impedance feature, Electrochemistry Communications, 98 (2019) 58-62.

[64] S. Diamond, S. Boyd, CVXPY: A Python-embedded modeling language for convex optimization, The Journal of Machine Learning Research, 17 (2016) 2909-2913.

[65] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, (2014).

[66] D. Maclaurin, D. Duvenaud, R.P. Adams, Autograd: effortless gradients in numpy, ICML 2015 AutoML workshop, 2015, pp. 5.

[67] G. Parascandolo, H. Huttunen, T. Virtanen, Taming the waves: sine as activation function in deep neural networks, (2016).

[68] H. Zheng, Z. Yang, W. Liu, J. Liang, Y. Li, Improving deep neural networks using softplus units, 2015 International Joint Conference on Neural Networks (IJCNN), IEEE, 2015, pp. 1-4.

[69] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), arXiv preprint arXiv:1511.07289, (2015).

[70] 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).

[71] E. Barsoukov, Impedance spectroscopy: theory, experiment, and applications, John Wiley 2017.

[72] N. Schlüter, S. Ernst, U. Schröder, Direct access to the optimal regularization parameter in distribution of relaxation times analysis, ChemElectroChem, 7 (2020) 3445-3458.

[73] 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.