GIANT Networks: Very Deep Fully Connected Neural Networks Applied to Microwave Problems

Downloaded from: https://research.chalmers.se, 2026-04-05 15:51 UTC

Citation for the original published paper (version of record):

Stenmark, S., Rylander, T., McKelvey, T. et al (2026). GIANT Networks: Very Deep Fully Connected Neural Networks Applied to Microwave Problems. IET Microwaves, Antennas and Propagation, 20(1). http://dx.doi.org/10.1049/mia2.70077

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

Check for updates

IET Microwaves, Antennas & Propagation

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

GIANT Networks: Very Deep Fully Connected Neural Networks Applied to Microwave Problems

Simon Stenmark | Thomas Rylander | Tomas McKelvey | Andrei Ludvig-Osipov

Department of Electrical Engineering, Chalmers University of Technology, Göteborg, Sweden

Correspondence: Tomas McKelvey (tomas.mckelvey@chalmers.se)

Received: 29 August 2025 | **Revised:** 28 November 2025 | **Accepted:** 13 December 2025

Keywords: microwave filters | microwave resonators | neural nets | optimisation | sensitivity analysis

ABSTRACT

We present the Gradient-Informed Attentive Normalisation Training (GIANT) framework with the objective to create very deep fully connected neural networks, which we use as surrogate models in the context of microwave problems. As the central component of the GIANT framework, we introduce a novel dynamic reparameterisation procedure for the weight-bias parameter space by means of a low-variance preserving normalisation layer for each fully connected layer and we refer to this construction as Attentive Normalisation (AttNorm). As part of AttNorm, we also introduce a new and tailored updating scheme that improves the convergence during training. To efficiently train very deep fully connected neural networks, we exploit Sobolev training with gradient information, which is computed at a very low computational cost by means of continuum sensitivity analysis. We test our novel approach on two microwave applications: (i) a six-port microwave cavity with a random medium and (ii) an H-plane waveguide filter optimised under geometrical uncertainty. For these examples, we demonstrate successful training of neural networks with up to 30 layers, which are sufficiently accurate and expressive to serve as excellent surrogate models.

1 | Introduction

The fast and accurate solution of Maxwell's equations is important in many microwave engineering applications that can be formulated as an optimisation problem or an inverse problem. In the parameter space for the sought quantities of the optimisation/inverse problem, it is often sufficient to consider a bounded region where Maxwell's equations typically must be solved a very large number of times in the search for a global solution to the optimisation/inverse problem at hand. Despite the very powerful computers of today and the very efficient computational methods developed in recent decades, it can be too expensive to work with a direct solution of Maxwell's equations by means of conventional methods such as the finite-difference time-domain scheme [1], the finite element method [2] or the method of moments [3]. Compared to full-wave solvers, an attractive alternative is to use a surrogate model [4], which is a computationally cheap model that is trained to emulate the full-wave solver in a limited region of the parameter space. Surrogate models allow for efficient optimisation of microwave components [5] where they can also be used in combination with space-mapping schemes [6–8].

Neural networks are attractive as surrogate models for microwave applications [5, 9, 10]. Neural networks can exhibit impressive flexibility that allows for the modelling of complicated nonlinear relationships between its input and output for relatively large regions in the input space. Additionally, once a neural network is ready to use, it can be evaluated very rapidly on a graphical processing unit (GPU). A number of different network architectures have emerged for the modelling of microwave components. Examples include neural networks that predict the poles and residues of pole-residue-based transfer

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *IET Microwaves, Antennas & Propagation* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

functions [11], knowledge-based neural networks that incorporate prior knowledge in the form of approximative analytical models of the components in their design [12, 13], and autoencoders that reduce the dimensionality of high-dimensional design parameter vectors [14]. As an important complement to such tailored solutions, fully connected neural networks are problem agnostic and can be applied without prior knowledge. Also, fully connected neural networks are often used as building blocks in more specialised neural network architectures. Finally, a common trend within the deep learning community is to repeatedly increase the depth of the neural network to model ever more complicated problems, which also naturally requires ever more data.

There are several approaches to improve the possibilities to train deeper neural networks. One approach is to use normalisation procedures such as BatchNorm [15]. However, the effectiveness of such procedures is application specific and depends upon parameters such as batch size and the choice of nonlinear activation function [16, 17]. In particular, BatchNorm has been noted to reduce performance for fully connected neural networks [18]. Another alternative is the natural neural network [19], which uses principal component analysis (PCA) whitening to enforce full whitening of the data for each internal layer of a fully connected neural network. In the context of microwave modelling, Jin et al. [20] use a custom architecture with a combination of sigmoid and ReLU activation functions and demonstrated networks with a depth of 14 layers. Another important aspect in the training of deep neural networks is the choice of weight optimiser where a proper choice of optimiser can drastically reduce the time needed to train the neural network [21]. Here, the Adam optimiser [22], which features an adaptive learning rate for each weight-bias parameter, is a popular and efficient choice.

Very deep and expressive networks require large amounts of training data, which are often expensive to collect and annotate. In the context of surrogate modelling, data are generated by computer simulations and each such data point can involve a large computational cost. Then, an attractive approach to increase the amount of information associated with each individual data point is to also include gradient information. Gradient information can be used in the training of the neural network [23–25] and such an approach belongs to the larger class of methods referred to as Sobolev training [26]. This approach is especially attractive for problems where the gradient information can be computed at a low computational cost.

In this paper, we present the Gradient-Informed Attentive Normalisation Training (GIANT) framework with the purpose to construct very deep fully connected neural networks that can act as surrogate models for full-wave numerical field solvers, given important classes of microwave problems. We propose a novel dynamic reparameterisation procedure for the weight-bias space that is based on a low-variance preserving normalisation layer for each fully connected layer. The low-variance preserving normalisation layers re-normalise the data between each fully connected hidden layer of the neural network where the objective is to maintain zero mean and a suitable standard deviation throughout the training process. This construction is referred to as Attentive Normalisation (AttNorm) and it also involves a new updating scheme that improves the convergence during training.

Given the objective to efficiently train very deep neural networks, we exploit Sobolev training for an objective function that involves the misfit of both function values and first-order derivatives. Sensitivity analysis offers a computationally efficient possibility to determine the first-order derivatives. An important family of problems concerns the prediction of scattering parameters as a function of the parameters that describe the material and shape of a device or scatterer where results on continuum sensitivity analysis [27–29] can be formulated in terms of the original field solution and an adjoint field solution. For reciprocal microwave circuits, the derivatives with respect to an unlimited number of material and shape parameters are available basically for free once the original field solutions have been computed for all the scattering parameters of the problem. Given fixed computational resources, Sobolev training and sensitivity analysis allow the efficient training of much deeper neural networks as compared to the more conventional approach that does not involve any derivatives of the output with respect to the input of the neural network.

We test the GIANT framework on two microwave problems: (i) a six-port cavity with a random medium that models a measurement system and (ii) an H-plane waveguide filter that is optimised under geometrical uncertainty. We demonstrate that the GIANT framework yields fully connected neural networks that give a substantially smaller error on an unseen test set compared to conventional fully connected neural networks.

2 | Method

In this section, we first describe our proposed dynamic reparameterisation procedure Attentive Normalisation (AttNorm). Also, we describe how gradient information can be used to enhance the training of very deep neural networks by means of Sobolev training.

2.1 | AttNorm Network Setup

Figure 1 shows schematically the basic architecture of the AttNorm network where features such as the number of layers and the width of different layers are tailored for the specific application at hand.

The AttNorm network predicts the output variables $\hat{y}_1, \hat{y}_2, ..., \hat{y}_P$ as a function of the input variables $x_1, x_2, ..., x_Q$. We organise the input variables into a vector $\mathbf{x}$ and the output variables into a vector $\hat{\mathbf{y}}$. Here and in the following, we use the hat symbol ($\hat{}$) to denote a quantity that is output by the neural network, and we denote the jth element of a vector $\mathbf{v}$ by v_j.

2.1.1 | Neural-Network Architecture

The AttNorm network contains L layers where each such layer is composed of two parts: (i) a conventional fully connected

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

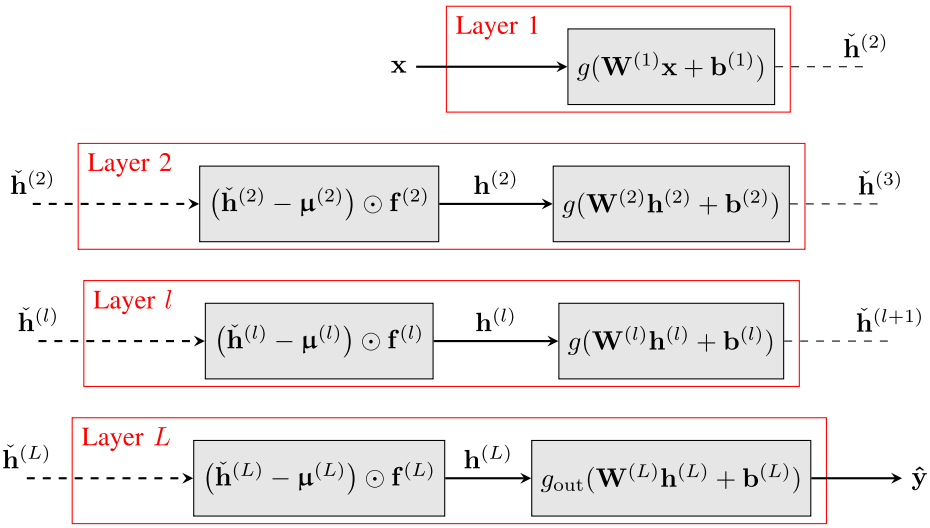

FIGURE 1 | Architecture of the neural network. We have the normalisation layers which normalise the input to each layer where the normalisation layer l is characterised by the sample mean values $\boldsymbol{\mu}^{(l)}$ and the scale factors $\mathbf{f}^{(l)}$. Here, the operator '$\odot$' denotes an element-wise product. The remaining layers are the fully connected layers where the fully connected layer l is characterised by the weight matrix $\mathbf{W}^{(l)}$ and the bias vector $\mathbf{b}^{(l)}$. Note that the output fully connected layer $l = L$ uses the activation function g_{out}, which is chosen for the application at hand.

layer and (ii) a normalisation layer. We use the super-index l to denote the lth layer where $l = 1, ..., L$.

2.1.2 | Fully Connected Layers

The lth fully connected layer is described by

$$\check{\mathbf{h}}^{(l+1)} = g\left(\mathbf{W}^{(l)}\mathbf{h}^{(l)} + \mathbf{b}^{(l)}\right) \tag{1}$$

where we have the weight matrix $\mathbf{W}^{(l)} \in \mathbb{R}^{N_l \times N_{l-1}}$ and the bias vector $\mathbf{b}^{(l)} \in \mathbb{R}^{N_l}$ for the number of neurons N_l in layer l. Further, we have the layer output $\check{\mathbf{h}}^{(l+1)} \in \mathbb{R}^{N_l}$ which, in turn, forms the input of layer $l + 1$. Here and in the following, we use the caron symbol (ˇ) to denote un-normalised quantities. The activation function is denoted $g(\mathbf{a})$ in Equation (1) and it applies element-wise to all elements of $\mathbf{a}$. For all numerical tests in this article, we use $g(\mathbf{a}) = \tanh(\mathbf{a})$.

2.1.3 | Normalisation Layers

We have the lth normalisation layer

$$h_i^{(l)} = \left(\check{h}_i^{(l)} - \mu_i^{(l)}\right) f_i^{(l)} \tag{2}$$

which centres and scales the input $\check{\mathbf{h}}^{(l)} \in \mathbb{R}^{N_{l-1}}$ of layer l before it is used in the activation function (1). Here, we have the offset parameters $\boldsymbol{\mu}^{(l)} \in \mathbb{R}^{N_{l-1}}$ and the scale factors $\mathbf{f}^{(l)} \in \mathbb{R}^{N_{l-1}}$.

2.1.4 | Input Layer

We assume that the input to the neural network is normalised. Thus, for the input layer $l = 1$, we omit the normalisation layer and we have $\mathbf{h}^{(1)} = \mathbf{x}$.

2.1.5 | Output Layer

For the output layer $l = L$, we have the activation function g_{out} which is chosen for the application at hand. For all numerical tests in this article, we use the linear activation function $g_{\text{out}}(\mathbf{a}) = \mathbf{a}$ and we have

$$\hat{\mathbf{y}} = \mathbf{W}^{(L)}\mathbf{h}^{(L)} + \mathbf{b}^{(L)}. \tag{3}$$

2.1.6 | Initialisation

We initialise all weight matrices $\mathbf{W}^{(l)}$ with the Glorot uniform initialiser [30], whereas all bias vectors $\mathbf{b}^{(l)}$ are initially set to zero.

For the normalisation layers, we initially assign $\mu_j^{(l)} = 0$ and $f_j^{(l)} = 1$ for all l and j. These values are then adjusted as part of the updating scheme, which is described in Section 2.3.

2.2 | Training, Validation and Test Sets

To optimise the weights of the neural network, we have a training set with the pairs $(\mathbf{x}^{\{d\}}, \mathbf{y}^{\{d\}})$, where $d = 1, ..., N_{\text{Tr}}$ is the

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

sample index and N_{Tr} is the number of samples in the training set. If we wish to employ Sobolev training, we assume that we also have access to the Jacobian matrix $\mathbf{J}$ with the entries

$$J_{pq} = \frac{\partial y_p}{\partial x_q} \tag{4}$$

for $p = 1, 2, ..., P$ and $q = 1, 2, ..., Q$ and instead form the triplets $\left(\mathbf{x}^{\{d\}}, \mathbf{y}^{\{d\}}, \mathbf{J}^{\{d\}}\right)$.

For the purpose of hyper-parameter tuning, we use a validation set with the number of samples N_{Val}. To verify the performance of the trained neural network, we use unseen data that we refer to as the test set where N_{Te} is the number of samples in the test set.

2.3 | AttNorm: Updating Scheme

We now wish to use the training set to optimise the parameters of the AttNorm network with the objective to minimise a loss function L where we present an example of a gradient-informed loss function in Section 2.5.2.

The updating scheme exploits the Adam optimiser [22] and involves three main steps that are repeated until the loss is sufficiently small. In the first step, we update the parameters of the normalisation layers and then adjust the weight-bias parameters of the fully connected layers such that the output of the neural network remains unchanged by the update. In the second step, we update the internal parameters of the Adam optimiser such that it remains consistent with the reparameterised weight space. In the third and final step, we apply the Adam optimiser to update the weight-bias parameters of the fully connected layers with the objective to minimise the loss function L.

2.3.1 | Update of Normalisation Parameters

In the first part of the updating scheme, the entire training set is passed through the network and the outputs $\mathbf{h}^{(l)}$ of all fully connected layers are stored. For simplicity, we consider a single layer l and omit the super-index (l) for all quantities in the remainder of this section.

First, we compute new values $\tilde{\mu}_j$ and $\tilde{f}_j$ for the offset parameters μ_j and scale factors f_j, respectively, according to

$$\tilde{\mu}_j = \frac{1}{N_{\text{Tr}}} \sum_{d=1}^{N_{\text{Tr}}} \check{h}_j^{\{d\}} \text{ and} \tag{5}$$

$$\tilde{f}_j = f(\tilde{\sigma}_j) \tag{6}$$

where we have the (un-normalised) output of the previous layer $\check{\mathbf{h}}^{\{d\}}$ that is associated with the input sample $\mathbf{x}^{\{d\}}$. Here, we use the scaling function $f(\sigma)$ to compute the scale factors based on the sample standard deviation

$$\tilde{\sigma}_j = \sqrt{\frac{1}{N_{\text{Tr}} - 1} \sum_{d=1}^{N_{\text{Tr}}} \left(\check{h}_j^{\{d\}} - \tilde{\mu}_j\right)^2}, \tag{7}$$

where we discuss the scaling function $f(\sigma)$ in Section 2.4.

Next, we modify the weight matrix $\mathbf{W}$ and the bias vector $\mathbf{b}$ to ensure that the argument of the activation function in (1) remains unchanged by the update given by the new values $\tilde{\mu}_j$ and $\tilde{f}_j$. In this process, we introduce the variables α_j and β_j such that the new values $\tilde{\mu}_j$ and $\tilde{f}_j$ are related to the current values μ_j and f_j according to

$$\alpha_j = \tilde{\mu}_j - \mu_j, \tag{8}$$

$$\beta_j = \frac{\tilde{f}_j}{f_j} \tag{9}$$

We now compute a new bias vector $\tilde{\mathbf{b}}$ according to

$$\tilde{b}_i = b_i + \sum_j W_{ij} f_j \alpha_j \tag{10}$$

and a new weight matrix $\tilde{\mathbf{W}}$ according to

$$\tilde{W}_{ij} = W_{ij}/\beta_j. \tag{11}$$

Finally, we replace μ_j, f_j, b_i and W_{ij} with their new counterparts $\tilde{\mu}_j$, $\tilde{f}_j$, $\tilde{b}_i$ and $\tilde{W}_{ij}$, which is applied for all layers of the neural network.

2.3.2 | Update of Adam Optimiser Parameters

In the second part of the updating scheme, we update the internal parameters of the Adam optimiser [22] such that it is consistent with the reparameterised weight space. For each weight-bias parameter θ, the Adam optimiser stores a momentum parameter $m\{\theta\}$ and a velocity parameter $v\{\theta\}$ where

$$m\{\theta\} \approx \mathbb{E}\left\{\frac{\partial L}{\partial \theta}\right\} \tag{12}$$

and

$$v\{\theta\} \approx \mathbb{E}\left\{\left(\frac{\partial L}{\partial \theta}\right)^2\right\} \tag{13}$$

for the loss function L. We use $\mathbb{E}\{x\}$ to denote the expected value of the random variable x where we consider the derivatives $\partial L/\partial\theta$ as independent random variables for the different θ. Here, the weight-bias parameter θ can be (i) an element of a bias vector $b_i^{(l)}$ or (ii) an element of a weight matrix $W_{ij}^{(l)}$. For simplicity, we once again consider a single layer l and thus omit the super-index (l) for the remainder of this section.

We first consider the case where θ is an element of a bias vector. From Equations (1) and (2), we find the input to the activation function

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$a_i = \sum_j W_{ij} f_j\left(h_j - \mu_j\right) + b_i. \tag{14}$$

from which we find

$$\frac{\partial L}{\partial b_i} = \frac{\partial L}{\partial a_i}\frac{\partial a_i}{\partial b_i} = \frac{\partial L}{\partial a_i}. \tag{15}$$

As all inputs to the activation functions a_i remain unchanged by the normalisation update procedure, we find that the momentum and velocity parameters that are associated with elements of bias vectors do not need to be updated.

Next, we consider the case where θ is an element of a weight matrix. We have

$$\frac{\partial L}{\partial W_{ij}} = \frac{\partial L}{\partial a_i}\frac{\partial a_i}{\partial W_{ij}} = \frac{\partial L}{\partial a_i} f_j\left(h_j - \mu_j\right). \tag{16}$$

After applying the update procedure (5–11), we find that

$$\begin{aligned}\frac{\partial L}{\partial \tilde{W}_{ij}} &= \frac{\partial L}{\partial a_i}\beta_j f_j\left(h_j - \left(\mu_j + \alpha_j\right)\right) \\ &= \beta_j \frac{\partial L}{\partial a_i} f_j\left(h_j - \mu_j\right) - \beta_j f_j \alpha_j \frac{\partial L}{\partial a_i} \\ &= \beta_j \frac{\partial L}{\partial W_{ij}} - \beta_j f_j \alpha_j \frac{\partial L}{\partial b_i}.\end{aligned} \tag{17}$$

Thus, for the optimiser parameters that are associated with an element of a weight matrix W_{ij} to be consistent with the update of the normalisation parameters, we update the corresponding momentum and velocity parameters as

$$m\left\{\tilde{W}_{ij}\right\} = \beta_j m\{W_{ij}\} - \beta_j f_j \alpha_j m\{b_i\} \tag{18}$$

and

$$\begin{aligned}v\left\{\tilde{W}_{ij}\right\} = &\left(\beta_j\right)^2 v\{W_{ij}\} \\ &- 2\left(\beta_j\right)^2 f_j \alpha_j m\{W_{ij}\} m\{b_i\} \\ &+ \left(\beta_j f_j \alpha_j\right)^2 v\{b_i\}.\end{aligned} \tag{19}$$

2.3.3 | Optimisation of Weight Matrices and Bias Vectors

In the third and final part of the updating scheme, we use the Adam optimiser [22] to update the weight matrices $\mathbf{W}^{(l)}$ and bias vectors $\mathbf{b}^{(l)}$ with the objective to minimise the loss function L for the entire neural network where the number of stochastic gradient steps executed before a new update of the normalisation parameters depend on the application.

2.4 | Low-Variance Preserving Scaling Function

In the update of the scale factors (6), we must choose a scaling function $f(\sigma)$ and, here, we wish to rescale the inputs to the fully connected layers of the neural network such that each input has a standard deviation that is close to unity. However, we assume that some of the variation of the inputs is noise and that nearly constant signals may be relatively more corrupted by such noise as compared to signals with a larger standard deviation. To avoid excessive amplification of noise associated with nearly constant signals, we hence propose a novel low-variance preserving (LVP) scaling function

$$f(\sigma) = \frac{\sigma + \epsilon}{\sigma^2 + \epsilon}. \tag{20}$$

here, we have the hyperparameter ϵ that must be chosen where we require $\epsilon \leq 1$ to achieve a low-variance preserving scaling function.

Given some typical values for ϵ in Equation (20), Figure 2 shows a comparison between two scaling functions: (i) solid lines—the scaling function $f(\sigma)$ and (ii) dashed lines—a conventional scaling function $f_{\text{conv}}(\sigma) = 1/\sqrt{\sigma^2 + \epsilon}$ that is used in normalisation techniques such as BatchNorm [15]. We note that the LVP scaling function Equation (20) reduces to $f(\sigma) \simeq 1 + \sigma/\epsilon$ for $\sigma^2 \ll \epsilon$, which further reduces to unit scaling $f(\sigma) \simeq 1$ for $\sigma \ll \epsilon$. In comparison, $f_{\text{conv}}(\sigma) \simeq 1/\sqrt{\epsilon}$ for $\sigma^2 \ll \epsilon$, which amplifies the standard deviation substantially for nearly constant signals of low variance given typical values for ϵ. For $\sigma^2 \gg \epsilon$ and $\epsilon \leq 1$, we have $f(\sigma) \simeq f_{\text{conv}}(\sigma)$ and the two scaling functions provide the same scaling. Finally, $f(\sigma)$ has its maximum at $\sigma = \sqrt{\epsilon(1+\epsilon)} - \epsilon$, which reduces to $\sigma = \sqrt{\epsilon}$ for $\epsilon \ll 1$. In the tests in this article, we find that $f(\sigma)$ performs better and is less sensitive to the choice of ϵ when compared to the conventional scaling $f_{\text{conv}}(\sigma)$.

2.5 | Sobolev Training

In this section, we introduce a gradient-informed loss function that allows for efficient Sobolev training of very deep fully connected neural networks.

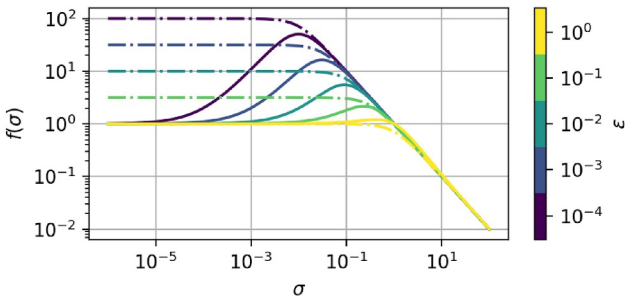

FIGURE 2 | The scaling function $f(\sigma)$ for five different values of the hyperparameter ϵ as a function of standard deviation σ compared to a conventional scaling function. Solid lines—scaling function $f(\sigma)$; and dashdotted lines—conventional scaling function $f_{\text{conv}}(\sigma) = 1/\sqrt{\sigma^2 + \epsilon}$.

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2.5.1 | Automatic Differentiation

We apply automatic differentiation to the neural network to compute the Jacobian matrix

$$\widehat{J}_{pq} = \frac{\partial \widehat{y}_p}{\partial x_q}. \tag{21}$$

where $\widehat{\mathbf{J}}$ is the Jacobian matrix as predicted by the neural network.

2.5.2 | Gradient-Informed Loss Function

We introduce the gradient-informed loss function

$$\begin{aligned} L = &\left[\frac{1}{BP}\sum_{b=1}^{B}\sum_{p=1}^{P}\left(\widehat{y}_p^{\{b\}} - y_p^{\{b\}}\right)^2\right] \\ &+ \alpha\left[\frac{1}{BPQ}\sum_{b=1}^{B}\sum_{p=1}^{P}\sum_{q=1}^{Q}\frac{1}{\sigma_{pq}^{(J)}}\left|\widehat{J}_{pq}^{\{b\}} - J_{pq}^{\{b\}}\right|\right] \end{aligned} \tag{22}$$

where the super-index b refers to the bth sample in the current batch of size B. The gradient-informed loss function (22) combines the mean-squared error of the function values with a penalisation term that involves the mean-absolute error of the derivatives (21) where α acts as a penalisation parameter. In the following, we consider α to be a constant that must be chosen by the user. The derivatives (4) and (21) may vary in magnitude significantly within a given data set despite a suitable normalisation of the input and/or output of the neural network. We mitigate such issues by weighting the derivatives in the loss function (22) with the inverse of the sample standard deviations $\sigma_{pq}^{(J)}$ computed from the element J_{pq} of the Jacobian matrix (4) given all the samples in the training set. It should be noted that the parameters $\sigma_{pq}^{(J)}$ are computed only once before the training starts. The user is advised to inspect the computed values since a small (or zero) value may indicate that the problem formulation should be reconsidered. Such issues do not occur in the examples studied in this article.

2.6 | The GIANT Network

To efficiently train very deep fully connected neural networks, we combine AttNorm (in Sections 2.1, 2.3 and 2.4) with the gradient-informed loss function (22), which we refer to as Gradient-Informed Attentive Normalisation Training (GIANT). Further, we refer to neural networks created by means of GIANT as GIANT networks.

The normalisation layer and fully connected layer can be combined by substituting $\mathbf{h}^{(l)}$ from (2) into the expression for the fully connected layer (1) to give the input to the activation function (14), which gives a combined weight matrix and bias vector. Thus, the evaluation of a trained GIANT network does not add any computational cost as compared to a conventional fully connected neural network.

3 | Modelling of Passive Microwave Circuits

We consider the modelling of a passive N_P-port microwave circuit. The circuit is characterised by its scattering parameters S_{ij} (with $i, j = 1, 2, ..., N_\mathrm{P}$) where the real and imaginary parts of the unique scattering parameters are organised in a real-valued (un-normalised) output vector $\check{\mathbf{y}}$ of length $P = N_\mathrm{P}(N_\mathrm{P} + 1)$ according to

$$\begin{aligned} \check{\mathbf{y}} = \Big[&\Re\{S_{11}\}, \Re\{S_{12}\}...\Re\{S_{1N_\mathrm{P}}\}, \\ &\Re\{S_{22}\}, \Re\{S_{23}\}...\Re\{S_{2N_\mathrm{P}}\}, \\ &..., \\ &\Re\{S_{N_\mathrm{P}N_\mathrm{P}}\}, \\ &\Im\{S_{11}\}, \Im\{S_{12}\}...\Im\{S_{1N_\mathrm{P}}\}, \\ &\Im\{S_{22}\}, \Im\{S_{23}\}...\Im\{S_{2N_\mathrm{P}}\} \\ &..., \\ &\Im\{S_{N_\mathrm{P}N_\mathrm{P}}\}\Big] \\ = &\left[\check{y}_1, \check{y}_2...\check{y}_P\right] \end{aligned} \tag{23}$$

We assume that the geometry and material parameters of the microwave circuit can be expressed by a set of parameters p_k (with $k = 1, 2, ..., Q - 1$). Thus, we form the (un-normalised) input vector

$$\check{\mathbf{x}} = \left[f, p_1, ..., p_{Q-1}\right] = \left[\check{x}_1, \check{x}_2, ..., \check{x}_Q\right]. \tag{24}$$

where the frequency f of the time-harmonic excitation is also included. An example of a parametric description of a complex permittivity distribution is given in Section 4.2.4, whereas an example of a parametric description of the geometry of a waveguide filter is given in Section 4.3.4.

3.1 | Boundary Value Problem

Given the input vector $\check{\mathbf{x}}$ that contains the excitation frequency and the description of the microwave circuit, we compute the scattering parameters in $\check{\mathbf{y}}$ based on the boundary value problem that models the microwave circuit. Two examples of boundary value problems are provided in the results section.

3.2 | Sensitivity Analysis

To be able to exploit Sobolev training, we now wish to compute the derivatives $\partial\check{y}_p/\partial\check{x}_q$.

For the derivative of the scattering parameters with respect to the frequency, we use the finite-difference approximation

$$\frac{\partial S_{ij}}{\partial f} \simeq \frac{S_{ij}\left(f + \Delta f, p_1, ..., p_{Q-1}\right) - S_{ij}\left(f, p_1, ..., p_{Q-1}\right)}{\Delta f} \tag{25}$$

with a suitable choice of Δf.

For p_k that describe the materials or geometry of microwave circuit, we use continuum sensitivity analysis [27–29] to compute the derivative $\partial S_{ij}/\partial p_k$ given the solution to the original field

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

problem and an adjoint field problem. For reciprocal problems, the adjoint field problem and the original field problem are identical, making the continuum sensitivities available at a negligible additional computational cost once all the scattering parameters of the N_P-port microwave circuit have been computed. An example of a material sensitivity is provided in Section 4.2, and an example of a shape sensitivity is provided in Section 4.3.

We wish to remark that although this paper considers reciprocal test-problems where the gradient information can be obtained for a very low computational cost, the GIANT framework can be applied to any problem where the derivatives of the outputs with regard to the inputs are available. However, the savings in terms of computational cost are dependent on the computational cost of obtaining the gradient information and need to be considered for each new problem of interest.

3.3 | Normalisation

Given $\check{\mathbf{x}}$ and $\check{\mathbf{y}}$, we compute the normalised input vector $\mathbf{x}$ and the normalised output vector $\mathbf{y}$ as

$$x_q = \frac{\check{x}_q - \mu_q^{(x)}}{\sigma_q^{(x)}} \tag{26}$$

$$y_p = \frac{\check{y}_p - \mu_p^{(y)}}{\sigma_p^{(y)}} \tag{27}$$

where the sample means $\mu_q^{(x)}, \mu_p^{(y)}$ and the sample standard deviations $\sigma_q^{(x)}, \sigma_p^{(y)}$ are computed from the samples in the training set.

Given Equations (26) and (27), we have the corresponding Jacobian matrix $\mathbf{J}$ associated with the normalised input and output variables

$$J_{pq} = \frac{\partial y_p}{\partial x_q} = \frac{\partial y_p}{\partial \check{y}_p}\frac{\partial \check{y}_p}{\partial \check{x}_q}\frac{\partial \check{x}_q}{\partial x_q} = \frac{\sigma_q^{(x)}}{\sigma_p^{(y)}}\frac{\partial \check{y}_p}{\partial \check{x}_q} \tag{28}$$

where the derivatives $\partial \check{y}_p / \partial \check{x}_q$ are computed as described in Section 3.2.

4 | Results

We demonstrate the capabilities of the suggested GIANT network on two reciprocal microwave problems: (i) modelling of a microwave measurement device intended for an inhomogeneous dielectric medium in a circular cavity and (ii) modelling of an H-plane filter with respect to its geometry with the application of optimising the reflection coefficient of the filter under geometrical uncertainty.

4.1 | Computer Implementation

In the numerical tests that follow, we use the Python framework TensorFlow [31] to implement the neural networks and to compute the Jacobian (21) by automatic differentiation.

To accommodate for training sets of varying sizes, we apply the Adam optimiser [22] for

$$n = \left\lceil \frac{750}{\text{Stochastic gradient - descent steps per epoch}} \right\rceil \tag{29}$$

full epochs before we execute the update of the normalisation parameters as described in Section 2.3 where an epoch corresponds to the stochastic gradient-descent steps required to traverse all the samples of the entire training set. This ensures that at least 750 gradient steps are performed before the normalisation parameters are updated where the constant 750 is empirically chosen to give a satisfactory performance for the AttNorm updating scheme.

We solve all boundary value problems with the FEM implemented in NGSolve [32]. We use quadratic node elements with a cell size chosen to achieve roughly 1% relative error. For the circular cavity in Test 1, this results in a maximum cell size of 0.9 mm. For the waveguide filter in Test 2, this results in a maximum cell size of 2 mm that is adaptively refined to 16 μm within the apertures of the filter. Similarly, all the derivatives based on sensitivity analysis for material and shape parameters are computed in NGSolve.

4.2 | Test 1—Stochastic Inhomogeneous Medium in Circular Cavity

The first example models a type of microwave measurement device intended for an inhomogeneous dielectric medium flowing through a metal pipe where Refs. [33, 34] provide further information. We use a random medium with an inhomogeneous permittivity to model the flowing dielectric medium where each new realisation of the random medium corresponds to a new snapshot of the medium that is currently inside the pipe. We wish to find a surrogate model that can predict the scattering parameters of the microwave circuit given (i) the excitation frequency f and (ii) a parametric description of the permittivity for the current realisation of the random medium.

4.2.1 | Geometry

The measurement device consists of a circular cavity connected to 6 parallel-plate waveguides as shown in Figure 3. The walls, shown by solid curves, are perfect electric conductors (PEC). This yields a cylindrical geometry where the cylinder axis is the z-axis. The diameter of the cavity is $D = 6$ cm and the width of the waveguides is 4.7 mm. The waveguides are terminated by waveguide ports, shown by dotted lines, at a distance of 3.8 cm from the centre of the circular cavity. The computational domain Ω is divided into two parts: (i) the six waveguides with a constant permittivity $\epsilon_{\text{wg}} = \epsilon_0 \epsilon_{\text{r,wg}}$ where $\epsilon_{\text{r,wg}} = 40$ and (ii) the circular cavity with a random medium that is characterised by the inhomogeneous permittivity $\epsilon_{\text{c}}(x, y) = \epsilon'_{\text{c}}(x, y) - \text{j}\epsilon''_{\text{c}}(x, y)$, which we further describe in Section 4.2.4.

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth> -member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

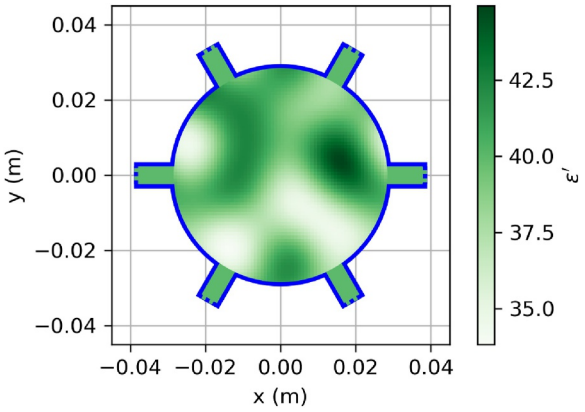

FIGURE 3 | Geometry for a model of a six-port measurement device connected to a circular cavity with a realisation of a random inhomogeneous permittivity where the colour corresponds to $\epsilon' = \epsilon'(x, y)$. The waveguide ports are shown by the dotted lines and the PEC surfaces by the solid curves.

4.2.2 | Boundary Value Problem

We solve for the magnetic field component $H_z = H_z(x, y)$ perpendicular to the computational domain Ω shown in Figure 3, which is enclosed by a combination of (i) the boundary Γ_N with a homogeneous Neumann boundary condition for the PEC walls and (ii) the boundary Γ_j for $j = 1, 2, ..., 6$ equipped with a Robin boundary condition for the jth waveguide port. Thus, we have the boundary value problem

$$-\nabla \cdot \left(\frac{1}{\epsilon_c} \nabla H_z\right) - \omega^2 \mu_0 H_z = 0 \quad \text{in } \Omega, \tag{30}$$

$$\hat{n} \cdot \nabla H_z = 0 \quad \text{on } \Gamma_N, \tag{31}$$

$$\hat{n} \cdot \nabla H_z + \gamma_0 H_z = 2\gamma_0 H_{0,j} \quad \text{on } \Gamma_j, \tag{32}$$

where $\gamma_0 = \mathrm{j}\sqrt{\epsilon_{r,wg}}\,\omega/c_0$ is the propagation constant of the fundamental transversal electromagnetic (TEM) mode in the waveguides. Here, $\hat{n}$ is the unit normal with respect to the boundary and it points away from the computational domain Ω. The jth port is excited with the fundamental waveguide mode of amplitude $H_{0,j}$. Given a nonzero excitation of the jth port only, the electric field amplitudes of the fundamental waveguide mode for the outward propagating waves are computed from $H_z(x, y)$ for all ports $i = 1, 2, ..., 6$ and, finally, the corresponding scattering parameters S_{ij} are computed.

4.2.3 | Sensitivity Analysis

Given the original field solution $H_{z,j}^{\mathrm{ori}} = H_{z,j}^{\mathrm{ori}}(x, y)$ with the jth port excited and the adjoint field solution $H_{z,i}^{\mathrm{adj}} = H_{z,i}^{\mathrm{adj}}(x, y)$ with the ith port excited, the sensitivity of the scattering parameters with respect to a perturbation of the permittivity $\delta\epsilon_c$ can be computed by

$$\delta S_{ij} = -\frac{\mu_0}{2a\gamma_0 E_{0,i}^{\mathrm{adj}} E_{0,j}^{\mathrm{ori}}} \int_\Omega \left[\frac{1}{\epsilon_c^2}\left(\nabla H_{z,i}^{\mathrm{adj}}\right) \cdot \left(\nabla H_{z,j}^{\mathrm{ori}}\right)\right] \delta\epsilon_c \, d\Omega, \tag{33}$$

where a is the width of all the waveguides. Furthermore, $E_{0,i}^{\mathrm{adj}}$ denotes the incident electric field amplitude associated with the ith port in the adjoint field problem. Similarly, $E_{0,j}^{\mathrm{ori}}$ is the incident electric field amplitude associated with the jth port in the original field problem. The derivatives with respect to the frequency f are computed by the finite-difference approximation (25) with $\Delta f = 10^{-6} f_0$ where the derivative is evaluated at the frequency $f = f_0$.

4.2.4 | Discrete Representation of the Permittivity

We expand the permittivity in the real-valued basis functions $\varphi_k(x, y)$ as

$$\epsilon_c(x, y) = \sum_{k=1}^{Q-1} \epsilon_{c,k} \varphi_k(x, y). \tag{34}$$

Here, we use the coefficients $\epsilon_{c,k} = \epsilon_0(1 - \mathrm{j}\tan\delta)p_k$ for $k = 1, ..., Q - 1$ where $\tan\delta$ is the loss tangent for the medium and we use $\tan\delta = 0.01$ in the tests that follow. To compute the derivatives $\partial S_{ij}/\partial p_k$, we exploit the sensitivity (33) with $\delta\epsilon_c$ as the perturbation of the permittivity that is associated with a perturbation of the coefficient p_k. The set of basis functions $\varphi_k(x, y)$ consists of the 2D Fourier basis functions with the spatial frequencies $\kappa_x = 2\pi\xi/L_x$ and $\kappa_y = 2\pi\zeta/L_y$ for integers ξ and ζ. We set $L_x = L_y = D$ and use all combinations of ξ and ζ such that $\sqrt{\xi^2 + \zeta^2} \le \sqrt{5}$, which results in a total of 21 independent basis functions with the highest spatial frequency $\kappa_{\max} = \sqrt{\kappa_x^2 + \kappa_y^2} = 2\pi\sqrt{5}/D$.

For the purpose of constructing a training set and a test set, we randomly generate $\left(p_1, p_2, ..., p_{Q-1}\right)$ to construct realisations of the random medium $\epsilon_c(x, y)$ such that the real part of the permittivity is in the interval $\left[\bar{\epsilon}' - \Delta\epsilon', \bar{\epsilon}' + \Delta\epsilon'\right]$. Here, we use $\bar{\epsilon}'/\epsilon_0 = 40$ and $\Delta\epsilon'/\epsilon_0 = 12$, which yields significant variations in the permittivity.

For each permittivity realisation of this random medium, we solve the boundary value problem (30–32) for 31 frequency points uniformly spaced from 1.5 to 3.0 GHz. Given these results (indexed by $k = 1, 2, ..., 31$ with respect to frequency), we form the sample-triplets $\left(\mathbf{x}^{\{k\}}, \mathbf{y}^{\{k\}}, \mathbf{J}^{\{k\}}\right)$ that are added to the training set and used independently of each other during the training. The size of the training set varies in the tests that follow and is specified when necessary. We use the same procedure to create a validation set that we use for hyperparameter tuning. The validation set is based on 500 permittivity realisations, which gives a total of $N_{\mathrm{Val}} = 15500$ samples. It should be emphasised that the mesh used for the FEM is fixed for all permittivity realisations, which facilitates comparisons of the computed scattering parameters and their sensitivities. To evaluate the performance of the trained neural networks, we construct a test set with the difference that the boundary value problem (30–32) is solved for 151 frequency points uniformly spaced from 1.5 to 3.0 GHz. We randomly generate 2000 permittivity realisations and, thus, we have in total $N_{\mathrm{Te}} = 302000$ samples in the test set.

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.2.5 | Choice of Hyperparameters

We use 150 neurons in each layer and initialise all weight matrices $\mathbf{W}^{(l)}$ with the Glorot uniform initialiser [30], whereas all bias vectors $\mathbf{b}^{(l)}$ are initially set to zero. In the loss function (22), we use $\alpha = 0.03$ to involve some gradient information in the loss function. During training, we use the learning rate 10^{-4} and the batch size 256. For the LVP scaling function (20), we use $\epsilon = 0.01$. All these hyperparameters have been tuned empirically to achieve a satisfactory performance of the trained neural network on the validation set and apply to all neural networks in this Section unless otherwise noted.

4.2.6 | Determination of Training Set Size

To evaluate the benefit of including gradient information in the training of a neural network, we use training sets of varying size to train a conventional fully connected neural network with 15 layers. Each network is trained for $1.2 \cdot 10^6$ gradient steps, which ensures adequate convergence for all training set sizes.

To measure the accuracy of the trained neural networks, we use the mean relative error (MRE) on the test set

$$\mathrm{MRE} = \frac{1}{N_{\mathrm{Te}}} \sum_{d=1}^{N_f} \frac{\|\widehat{\mathbf{S}}^{\{d\}} - \mathbf{S}^{\{d\}}\|_{\mathrm{F}}}{\|\mathbf{S}^{\{d\}}\|_{\mathrm{F}}}, \tag{35}$$

where we have the scattering matrix $\widehat{\mathbf{S}}$ as predicted by the neural network and the scattering matrix $\mathbf{S}$ from the FEM solver, which we consider the correct solution. Also, we use $\|\mathbf{S}\|_{\mathrm{F}}$ to denote the Frobenius norm of the matrix $\mathbf{S}$. It should be noted that the scattering matrices $\widehat{\mathbf{S}}^{\{d\}}$ are computed from the output of the neural network $\widehat{\mathbf{y}}^{\{d\}}$ as described by (23) and (27). Figure 4 shows the MRE as a function of the number of samples N_{Tr} in the training set for two cases: (i) blue solid curve—the loss function (22) involves both the function values and their derivatives with $\alpha = 0.03$ and (ii) red dashed curve—the loss function (22) involves only the function values and $\alpha = 0$. We note that the incorporation of the gradient information drastically decreases the number of samples needed to achieve a given MRE where we emphasise that the gradient information is provided (basically) for free given a reciprocal N_{P}-port microwave device. In fact, Figure 4 displays that the low MRE of the GIANT network cannot be achieved if the loss function only involves the function values ($\alpha = 0$) given the size of training set used here. Thus, we conclude that the incorporation of the gradient information in the training of the neural network yields significant computational savings.

4.2.7 | Evaluation of Normalisation Methods From Literature

For comparison with the open literature, we train three different fully connected neural networks with 20 layers each: (i) a neural network that does not exploit any normalisation, (ii) a neural network equipped with BatchNorm layers [15] between each fully connected layer; and (iii) a natural neural network [19].

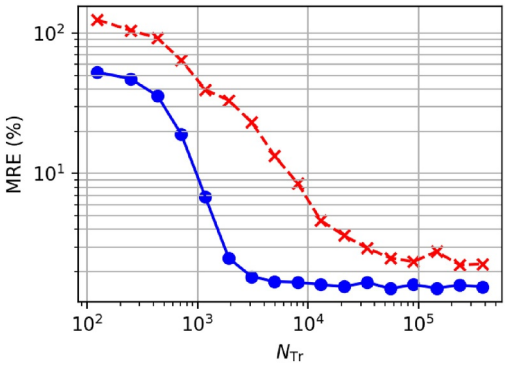

FIGURE 4 | MRE on the test set as a function of training set size N_{Tr}: blue solid curve—the loss function (22) involves both the function values and their derivatives with $\alpha = 0.03$ and red dashed curve—the loss function (22) involves only the function values and $\alpha = 0$.

The conventional neural network and the neural network with BatchNorm layers use the Adam optimiser [22], whereas the natural neural network uses the optimisation scheme that is described by Desjardins et al. [19]. To achieve a fair comparison, all networks are trained based on the loss function in Equation (22) with $\alpha = 0.03$. The conventional neural network and the neural network with BatchNorm layers use the learning rate 10^{-4} whereas the natural neural network uses the learning rate 10^{-3} where the hyperparameter ϵ is set to 10^{-3} for both the neural network with BatchNorm layers and the natural neural network. Here, these hyperparameters are tuned empirically to achieve the best possible training performance. Figure 5 shows a representative example of the training loss as a function of training epoch for the three types of networks: blue solid line—neural network without any normalisation, red dashed line—neural network equipped with BatchNorm layers and green dashdotted line—natural neural network. We see that both the neural network that is equipped with BatchNorm layers and the natural neural network are outperformed by the neural network without any normalisation. Thus, we use standard fully connected neural networks without any normalisation as the reference for the GIANT networks.

It is noteworthy that both the neural network equipped with BatchNorm layers and the natural neural network are outperformed by the conventional neural network. Klambauer et al. [18] note a decrease in training performance when they equip fully connected neural networks with BatchNorm layers, which they attribute to perturbations of the training process that are associated with the updates of the normalisation parameters. In this regard, there are several key differences between AttNorm and BatchNorm. Although BatchNorm updates the normalisation parameters for every mini-batch based on the samples of the current batch, AttNorm uses the whole training set to estimate the normalisation parameters at an interval which is set by the user. Additionally, the output of a neural network that is equipped with BatchNorm layers changes when the normalisation parameters are updated, whereas AttNorm adjusts the weight matrices and bias vectors of the neural network such that the output of the neural network remains unchanged by the update. The natural neural network is similar to AttNorm in that the output of the natural neural network

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

remains unchanged by the update of the normalisation parameters. However, the natural neural network uses basic stochastic gradient descent (SGD) to optimise the weight matrices and bias vectors between the updates of the normalisation parameters, whereas AttNorm is designed to work in tandem with the Adam optimiser [22], which can drastically improve convergence when compared to basic SGD. Another major difference is that the natural neural network estimates full covariance matrices to enforce full whitening of the data between the layers of the neural network, whereas AttNorm only attempts to normalise the variances of the data.

4.2.8 | Evaluation of AttNorm Performance

First, we compare the performance of the LVP scaling function (20) to a conventional scaling function $f_{\text{conv}}(\sigma) = 1/\sqrt{\sigma^2 + \epsilon}$. To this extent, we train several 20-layer neural networks with both types of scaling functions for different values of the hyperparameter ϵ. We find that (i) for both scaling functions, the final training loss increases if the hyperparameter ϵ is set to $5 \cdot 10^{-2}$ or above and (ii) the training process becomes unstable and the training loss diverges if it is set to $4 \cdot 10^{-3}$ or below for the conventional scaling function and to $6 \cdot 10^{-4}$ or below for the LVP scaling function. We note that the proper choice of ϵ for a stable training process is dependent on other hyperparameters such as learning rate and the architecture of the neural network. In our experience, the LVP scaling function (20) is not very sensitive to the choice of the hyperparameter ϵ and provides a robust and stable training process for a wider range of hyperparameters than the conventional scaling function. Out of the tested combinations, we find that the best training performance is achieved with the LVP scaling function with $\epsilon = 0.01$ and we use this combination for the remainder of the article.

Finally, we evaluate the performance of the full GIANT framework. We train several GIANT networks and conventional neural networks for an increasing number of layers. All networks are trained for 600 epochs with a learning rate of 10^{-4}, then fine-tuned for 10 epochs with a learning rate of 10^{-5}. Here, the training set contains 12,300 permittivity realisations, which gives in total $N_{\text{Tr}} = 381300$ samples. Given the training loss (22) with $\alpha = 0.03$, Figure 6 shows the training loss as a function of training epoch for two networks with 20 layers each: (i) blue solid line—GIANT network and (ii) red dashed line—conventional neural network. Note the sudden decrease in the training loss during the last 10 training epochs, which shows the effect of the fine-tuning procedure where the learning rate is decreased during the last few epochs. We see that the rate of convergence is significantly improved for the GIANT network where the conventional neural network requires at least double the number of training epochs to reach a similar performance in terms of the training loss.

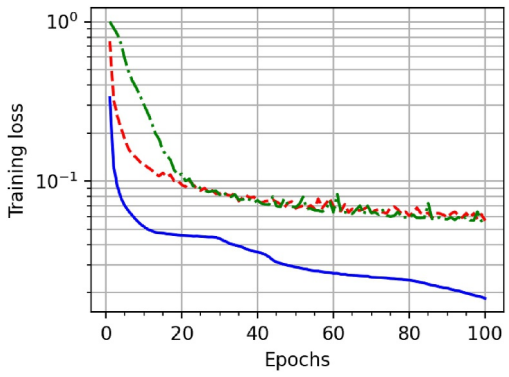

FIGURE 5 | Training loss as a function of epochs for three types of neural network: blue solid line—neural network without any normalisation, red dashed line—neural network equipped with BatchNorm layers and green dashdotted line—natural neural network.

Figure 6 involves 20 hidden layers and next, we study the effect of varying the number of layers in this context. For this purpose, Figure 7 compares the MRE (35) on the test set for the two trained architectures: (i) red crosses and dashed curve—conventional neural networks and (ii) blue disks and solid curve—the GIANT network proposed in this article. (The different glyphs correspond to different neural networks based on different randomly generated initial weights where the curves show the average MRE given these realisations.) We find that the GIANT

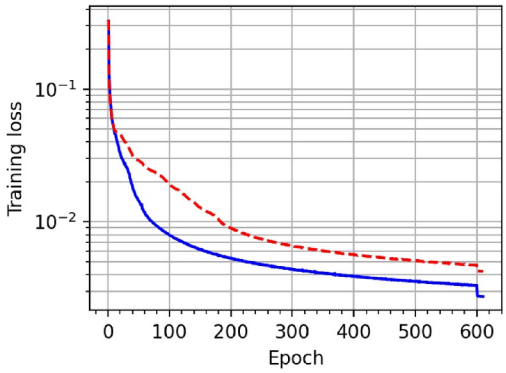

FIGURE 6 | Training loss as a function of training epoch for two neural networks with 20 layers: blue solid line—GIANT network and red dashed line—conventional neural network.

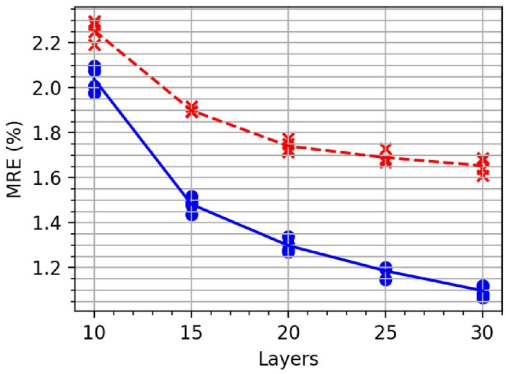

FIGURE 7 | MRE on the test set as a function of number of hidden layers. Blue solid curve with blue discs—GIANT networks and red dashed curve with red crosses—conventional neural networks. Each glyph (cross or disc) indicates a separate neural network with randomly initialised weights. For each number of hidden layers, we train four of each type of network. The curves consist of straight lines that connect the average MRE for each number of hidden layers.

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

networks yield a significantly lower MRE as compared to the conventional neural network for all network depths tested where the difference increases with the number of layers.

4.2.9 | Performance of Trained GIANT Network

We now wish to evaluate the feasibility of employing a GIANT network as a computationally efficient surrogate model for the field solver. To this extent, we evaluate the performance of a 15-layer GIANT network trained with a training set that contains $N_{\mathrm{Tr}} = 1922$ samples, which is based on Figure 4 that shows that the performance of the GIANT network is only marginally improved by further increasing the size of the training set.

For each of the 2000 permittivity realisations in the test set, we compute the relative error

$$\mathrm{RE}(\epsilon_c) = \frac{1}{N_f} \sum_{k=1}^{N_f} \frac{\|\widehat{\mathbf{S}}^{\{k\}}(\epsilon_c) - \mathbf{S}^{\{k\}}(\epsilon_c)\|_{\mathrm{F}}}{\|\mathbf{S}^{\{k\}}(\epsilon_c)\|_{\mathrm{F}}}, \tag{36}$$

where the super-index k refers to the kth frequency point for the current permittivity realisation ϵ_c with the total number of frequency points N_f.

Figure 8 shows the cumulative frequency of the RE (36) for all permittivity realisations in the test set given the scattering matrices as predicted by the GIANT network. We note that a large majority of the 2000 permittivity realisations have an RE that is below 5%. The MRE (35) for the test set is 2.5% with a maximum RE of 19.2%.

The blue solid curve in Figure 9 shows $|\widehat{S}_{14}(f)|$ predicted by the GIANT network for the permittivity realisation with the highest RE in the test set. In addition, the green dashed curve in Figure 9 shows $|S_{14}(f)|$ from the FEM solver, which is considered to be the correct result. We note that $|\widehat{S}_{14}(f)|$ is in reasonable agreement with $|S_{14}(f)|$ from the FEM solver.

For comparison with the conventional neural network training approach, we also train a conventional neural network on the same training set where the loss function (22) involves only the function values and $\alpha = 0$.

Figure 10 shows the cumulative frequency of the RE for all permittivity realisations in the test set given the scattering matrices as predicted by the conventional neural network. We note that a majority of the permittivity realisations have an RE that is above 20%. The MRE for the test set is 30.9% with a maximum RE of 107%.

The blue solid curve in Figure 11 shows $|\widehat{S}_{14}(f)|$ as predicted by the conventional neural network for the permittivity realisation with the highest RE in the test set, whereas the green dashed curve shows $|S_{14}(f)|$ from the FEM solver. Here, $|\widehat{S}_{14}(f)|$ as predicted by the conventional neural network shows little to no resemblance to $|S_{14}(f)|$ from the FEM solver.

In conclusion, we find that the GIANT framework allows the construction of a surrogate model of acceptable accuracy with a training set with as few as 1922 samples (which corresponds to 62 permittivity realisations) where the accuracy can be further improved by increasing the size of the training set. For the conventional neural network training approach, a similarly sized training set is clearly insufficient to yield an acceptable surrogate model where Figure 4 indicates that a substantially

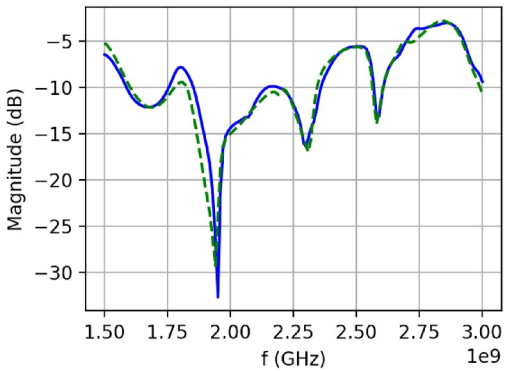

FIGURE 9 | A representative scattering parameter for the permittivity realisation with the maximum RE of the test set for the GIANT network. Blue solid line—$|\widehat{S}_{14}(f)|$ as predicted by the GIANT network and green dashed line—$|S_{14}(f)|$ from the FEM solver.

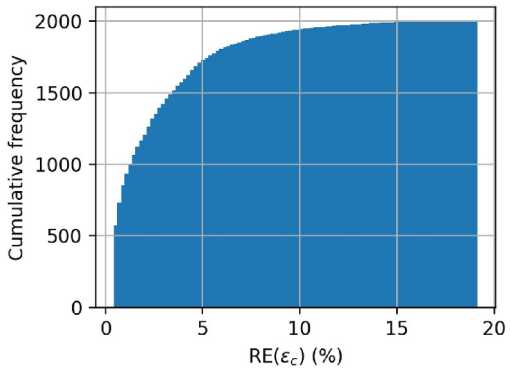

FIGURE 8 | Cumulative frequency of the RE for all 2000 permittivity realisation in the test set for a GIANT network trained on a training set with 1922 samples.

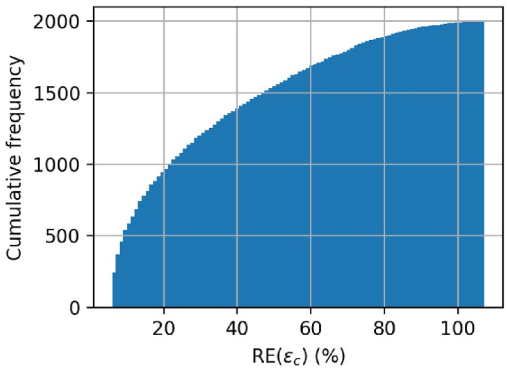

FIGURE 10 | Cumulative frequency of the RE for all 2000 permittivity realisation in the test set for a conventional neural network trained on a training set with 1922 samples.

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

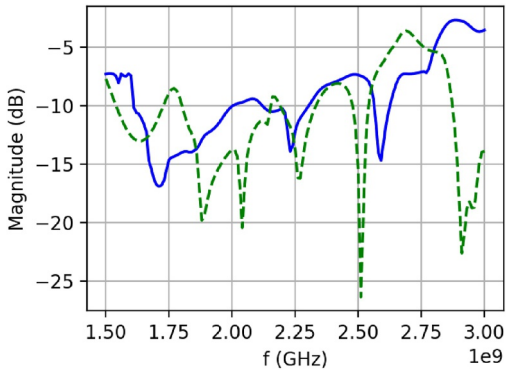

FIGURE 11 | A representative scattering parameter for the permittivity realisation with the maximum MRE of the test set for the conventional neural network. Blue solid line—$|\widehat{S}_{14}(f)|$ as predicted by the conventional neural network and green dashed line—$|S_{14}(f)|$ from the FEM solver.

larger training set is required to construct a surrogate model of comparable accuracy.

4.2.10 | Example Application—Multidimensional Histograms

Given the GIANT network trained and tested above, we now generate data points to populate histograms in a high-dimensional space, which is a computationally demanding problem that greatly benefits from learning a representation of Maxwell's equations that can be sampled on a massive scale by means of GPUs. Such histograms can, for example, be used to construct efficient algorithms for detection problems [33].

At the frequency $f = 2.5$ GHz, we consider a four-dimensional histogram with respect to $\Re\{S_{11}\}$, $\Im\{S_{11}\}$, $\Re\{S_{12}\}$ and $\Im\{S_{12}\}$ where the remaining scattering parameters are marginalised out. We use 20 bins along each dimension of the histogram, in total $20^4 = 160000$ bins, and we wish to construct it based on a number of samples that is substantially larger than the number of bins to achieve a histogram of acceptable resolution. Figure 12 shows a slice of this four-dimensional histogram based on 6,553,600 samples estimated by the GIANT network where $0.315 < \Re\{S_{12}\} < 0.325$ and $0.124 < \Im\{S_{12}\} < 0.134$. On a consumer-grade PC equipped with a Nvidia GeForce RTX 2080 Ti GPU, it takes approximately 1 s to create a sample using the FEM solver while it takes approximately 1 microsecond to create a sample using the GIANT network. To give an overall account of the computational time, we conclude that the creation of the data sets, training of the GIANT network and its execution to evaluate the scattering parameters (say at 31 frequency points) for 6,553,600 realisations of the random medium takes in total about 6 h. As a contrasting example, the direct evaluation by means of the FEM solver with the objective to compute the scattering parameters at 31 frequency points for 6,553,600 realisations of the random medium yields a total computational time of more than 6 years.

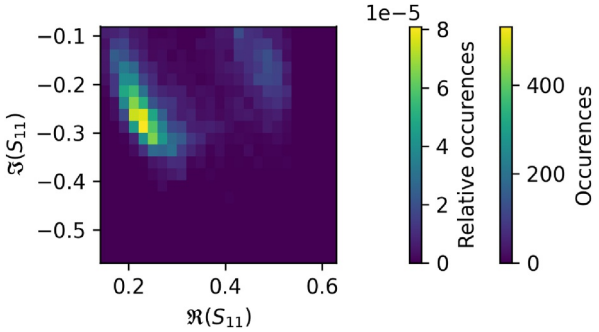

FIGURE 12 | A slice of a 4D histogram for ($\Re\{S_{11}\}$, $\Im\{S_{11}\}$, $\Re\{S_{12}\}$, $\Im\{S_{12}\}$) at $f = 2.5$ GHz as a function of ($\Re\{S_{11}\}$, $\Im\{S_{11}\}$) estimated from 6,553,600 samples computed by the GIANT network.

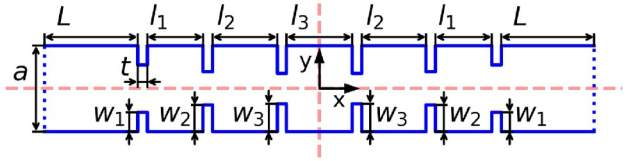

FIGURE 13 | Geometry of the H-plane waveguide filter. The dimensions w_1, w_2, w_3, l_1, l_2 and l_3 are subject for optimisation. The remaining parameters $a = 22.86$ mm, $t = 2.54$ mm and $L = 25$ mm are fixed. The waveguide ports are shown by the dotted lines and the PEC surfaces by the solid curves.

4.3 | Test 2—Optimised H-Plane Filter Under Uncertainty

We now consider the H-plane waveguide filter shown in Figure 13. Here, we wish to find a surrogate model that can predict the scattering parameters of the filter given (i) the excitation frequency f and (ii) a parametric description of the geometry of the filter.

4.3.1 | Geometry

The geometry features five cavities and it has two symmetry planes shown by the dashed horizontal line, that is, the x-axis and the dashed vertical line, that is, the y-axis. The width of the waveguide is $a = 22.86$ mm and the medium inside the waveguide is air, which gives the cut-off frequency 6.56 GHz for the fundamental mode TE_{10}. The width of the walls that separate the cavities is $t = 2.54$ mm. The distance from the cavities to the ports is $L = 25$ mm.

4.3.2 | Boundary Value Problem

We solve for the electric field component $E_z(x, y)$ perpendicular to the computational domain Ω shown in Figure 13, which is enclosed by a combination of (i) the boundary Γ_D with a homogeneous Dirichlet boundary condition for the PEC walls shown by solid curves and (ii) the boundary Γ_j for $j = 1, 2$ equipped with a Robin boundary condition for the jth waveguide port, shown by dotted lines. Thus, we have the boundary value problem

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$-\nabla\cdot\left(\frac{1}{\mu_0}\nabla E_z\right)-\omega^2\epsilon_0 E_z=0 \qquad \text{in } \Omega, \tag{37}$$

$$E_z=0 \qquad \text{on } \Gamma_D, \tag{38}$$

$$\hat{n}\cdot\nabla E_z+\gamma_{10}E_z=2\gamma_{10}E_{z,j}^{\text{inc}} \qquad \text{on } \Gamma_j, \tag{39}$$

where the propagation constant is $\gamma_{10}=\mathrm{j}\sqrt{(\omega/c_0)^2-(\pi/a)^2}$ and the incident field is $E_{z,j}^{\text{inc}}=E_{0,j}\sin(\pi y/a)$ for the fundamental TE_{10} mode. Here, $\hat{n}$ is the unit normal with respect to the boundary and it points away from the computational domain Ω. Given the field solution $E_z(x,y)$, we compute the scattering parameters S_{11} and S_{21}.

4.3.3 | Sensitivity Analysis

Given the original field solution $E_{z,j}^{\text{ori}}=E_{z,j}^{\text{ori}}(x,y)$ with the jth port excited and the adjoint field solution $E_{z,i}^{\text{adj}}=E_{z,i}^{\text{adj}}(x,y)$ with the ith port excited, the sensitivity of the scattering parameters with respect to a normal displacement δn of the boundary Γ_D (without perturbing the waveguide ports) can be computed from

$$\delta S_{ij}=\frac{1}{a\gamma_{10}E_{0,i}^{\text{adj}}E_{0,j}^{\text{ori}}}\int_{\Gamma_D}\left[\nabla E_{z,i}^{\text{adj}}\cdot\nabla E_{z,j}^{\text{ori}}\right]\delta n\, d\Gamma \tag{40}$$

where $E_{0,i}^{\text{adj}}$ is the amplitude of the incident electric field that excites port i in the adjoint field problem and $E_{0,j}^{\text{ori}}$ is the incident electric field amplitude that excites port j in the original problem. The derivatives with respect to the frequency f are computed by the finite-difference approximation (25) with $\Delta f=10^{-6}f_0$ where the derivative is evaluated at the frequency $f=f_0$.

4.3.4 | Discrete Representation of the Geometry

The geometry parameters that are listed in $\check{\mathbf{x}}$ are $p_1=w_1$, $p_2=w_2$, $p_3=w_3$, $p_4=l_1$, $p_5=l_2$ and $p_6=l_3$. To compute the derivatives $\partial S_{ij}/\partial p_k$, we exploit the sensitivity (40) with δn as the normal displacement of the boundary Γ that is associated with a perturbation of the parameter p_k.

To generate training data and test data, we use uniform random distributions to draw realisations of the geometry g, where

$$w_i\sim U(\overline{w}_i-\Delta w,\overline{w}_i+\Delta w) \tag{41}$$

$$l_i\sim U(\overline{l}_i-\Delta l,\overline{l}_i+\Delta l) \tag{42}$$

for $i=1,2$ and 3. Here, we use $\Delta w=1$ mm and $\Delta l=2$ mm. The average filter geometry $\overline{g}$ is described by the parameters shown in Table 1 and is chosen to yield a filter with reasonable pass-band filter characteristics.

We randomly generate 10,000 geometries that are used to construct a training set. For each of these geometry realisations, we solve the boundary value problem (37–39) for 21 frequency points uniformly spaced from $f_{\min}=9.5$ GHz to $f_{\max}=10.5$ GHz. Given these results, we create the sample-triplets $(\mathbf{x}^{\{k\}},\mathbf{y}^{\{k\}},\mathbf{J}^{\{k\}})$ that form the training set and are used independently of each other during the training. Thus, the training set involves $N_{\text{Tr}}=210000$ samples.

TABLE 1 | Average filter geometry.

$\overline{w}_1$ (mm)	$\overline{w}_2$ (mm)	$\overline{w}_3$ (mm)	$\overline{l}_1$ (mm)	$\overline{l}_2$ (mm)	$\overline{l}_3$ (mm)
5.1	7.0	7.4	15.1	17.3	17.6

For tuning of hyper-parameters, a validation set is constructed in the corresponding manner based on 1000 randomly generated geometries and 99 uniformly distributed frequency points in the interval from $f_{\min}=9.51$ GHz to $f_{\max}=10.49$ GHz. Thus, the validation set contains $N_{\text{Val}}=99000$ samples in total.

4.3.5 | Training and Evaluation of GIANT Network

In the following, we use 20 hidden layers, where each layer features 150 neurons. The loss function (22) has the weight $\alpha=0.05$ for the gradient information. During training, we use the batch size 256. In addition, we use a two-step learning rate schedule, where we train the GIANT network for a total of 2500 training epochs. During the first 1000 epochs, we use the learning rate 10^{-5} and then we reduce the learning rate to 10^{-6} for the remaining the final 1500 epochs. For the LVP scaling function (20), we use $\epsilon=0.01$. All these hyperparameters are a result of empirical tuning to achieve a satisfactory performance of the trained GIANT network.

4.3.6 | Example Application—Geometry Optimisation Under Uncertainty

As an application for the trained GIANT network that models the H-plane waveguide filter, we optimise the filter geometry described by $\mathbf{p}$ to obtain a band-pass filter. Here, we have the pass band $\left[f_{\text{pb,min}},f_{\text{pb,max}}\right]$ where we wish to minimise the maximum value of $|S_{11}|$. The stop band occurs in the two frequency bands $[f_{\min},f_{\text{sb,max}}]$ and $[f_{\text{sb,min}},f_{\max}]$ where $|S_{11}|$ must exceed the threshold $\Delta_{\text{sb}}=-3$ dB. We formulate this as the constrained optimisation problem

$$\begin{aligned}\min_{\mathbf{p}}\quad & \max_{f\in\left[f_{\text{pb,min}},f_{\text{pb,max}}\right]} 20\log_{10}(|S_{11}(f;\mathbf{p})|)\\ \text{s.t.}\quad & 20\log_{10}(|S_{11}(f;\mathbf{p})|)\geq\Delta_{\text{sb}} \quad \text{for } f\in\left[f_{\min},f_{\text{sb,max}}\right]\\ & 20\log_{10}(|S_{11}(f;\mathbf{p})|)\geq\Delta_{\text{sb}} \quad \text{for } f\in\left[f_{\text{sb,min}},f_{\max}\right]\end{aligned} \tag{43}$$

where $f_{\min}<f_{\text{sb,max}}<f_{\text{pb,min}}<f_{\text{pb,max}}<f_{\text{sb,min}}<f_{\max}$. We define the bandwidth $B=f_{\text{pb,max}}-f_{\text{pb,min}}$ and choose

$$\begin{aligned}f_{\text{sb,max}}&=f_{\text{mid}}-B/2-\Delta f,\\ f_{\text{pb,min}}&=f_{\text{mid}}-B/2,\\ f_{\text{pb,max}}&=f_{\text{mid}}+B/2,\\ f_{\text{sb,min}}&=f_{\text{mid}}+B/2+\Delta f\end{aligned}$$

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where $\Delta f = 0.1$ GHz and $f_{\text{mid}} = (f_{\max} + f_{\min})/2$ with $f_{\min} = 9.5$ GHz and $f_{\max} = 10.5$ GHz. In the following, we vary the bandwidth B from $B_{\min} = 0.2$ GHz to $B_{\max} = 0.6$ GHz.

Next, we use a brute-force approach to solve the optimisation problem (43) by means of straight-forward grid search, which is limited to a smaller and smaller region as a minimum is approached. Here, we augment the training data set adaptively with a small number of data points in the vicinity of the minimum as it is approached, which is combined with fine tuning of the neural network for improved convergence properties of the optimisation method. Figure 14 shows the scattering parameter $|S_{11}|$ as a function of frequency for three different bandwidths (a) $B = 0.2$ GHz, (b) $B = 0.4$ GHz and (c) $B = 0.6$ GHz where the pass and stop bands are indicated by the red areas. For each bandwidth, we include three different scattering parameter results: (i) solid curve—$|\widehat{S}_{11}|$ as predicted by the GIANT network for the optimised geometry $\mathbf{p}_{\text{opt}}$, (ii) dashed curve—$|S_{11}|$ computed by the FEM for the optimised geometry $\mathbf{p}_{\text{opt}}$, (iii) dash-dotted curve—the error $|\widehat{S}_{11} - S_{11}|$ for the GIANT network when compared to the FEM and (iv) dotted curve—$|S_{11}|$ computed by the FEM for the best available geometry in the training set and the test set according to the optimisation problem (43). It is clear that the best sample in the training and validation sets yields a sub-optimal solution in comparison with $|\widehat{S}_{11}|$ for $\mathbf{p}_{\text{opt}}$ provided by the trained GIANT network. A comparison of $|\widehat{S}_{11}|$ for $\mathbf{p}_{\text{opt}}$ with the corresponding result computed by the FEM validates that the GIANT network indeed yields accurate estimates also for optimised geometries. It is clear from Figure 14 that as the bandwidth B is increased, the maximum reflection coefficient in the pass band for the optimised designs increases: (i) $B = 0.2$ GHz yields -44.5 dB, (ii) $B = 0.4$ GHz yields -32.7 dB and (iii) $B = 0.6$ GHz yields -25.2 dB.

The optimised geometry parameters are presented in Table 2. We note that these geometry parameters are very sensitive to perturbations.

Finally, we explore optimisation under uncertainty. Apart from a design $\mathbf{p}$, we explore the geometry parameters $\mathbf{p}$ subject to all possible combinations of perturbations -10 μm, 0 and $+10$ μm. This yields a total of $3^6 = 729$ geometry assessments with the unperturbed case included and we require that all these designs fulfil the optimization problem in order to accept $\mathbf{p}$ as an optimised solution of (43). Figure 15 shows the scattering parameter $|S_{11}|$ as a function of frequency for three different bandwidths (a) $B = 0.2$ GHz, (b) $B = 0.4$ GHz and (c) $B = 0.6$ GHz where the pass and stop bands are indicated by the red areas. Again, we find that the GIANT network provides significant added value as compared to the best sample in the validation sets. Also, the scattering parameters for the optimised design predicted by the GIANT network is in good agreement with the corresponding result computed by the FEM. It should be noted that the gap between the masks (shown by the red areas) and optimised designs is necessary to fulfil the optimization problem with the geometrical perturbations. Figure 15 also shows that as the bandwidth B is increased, the maximum reflection coefficient in the pass band for the optimised designs increases. However, the perturbation of the geometry parameters makes this effect less pronounced: (i) $B = 0.2$ GHz yields -27.1 dB, (ii) $B = 0.4$ GHz yields -26.0 dB, and (iii) $B = 0.6$ GHz yields -21.9 dB.

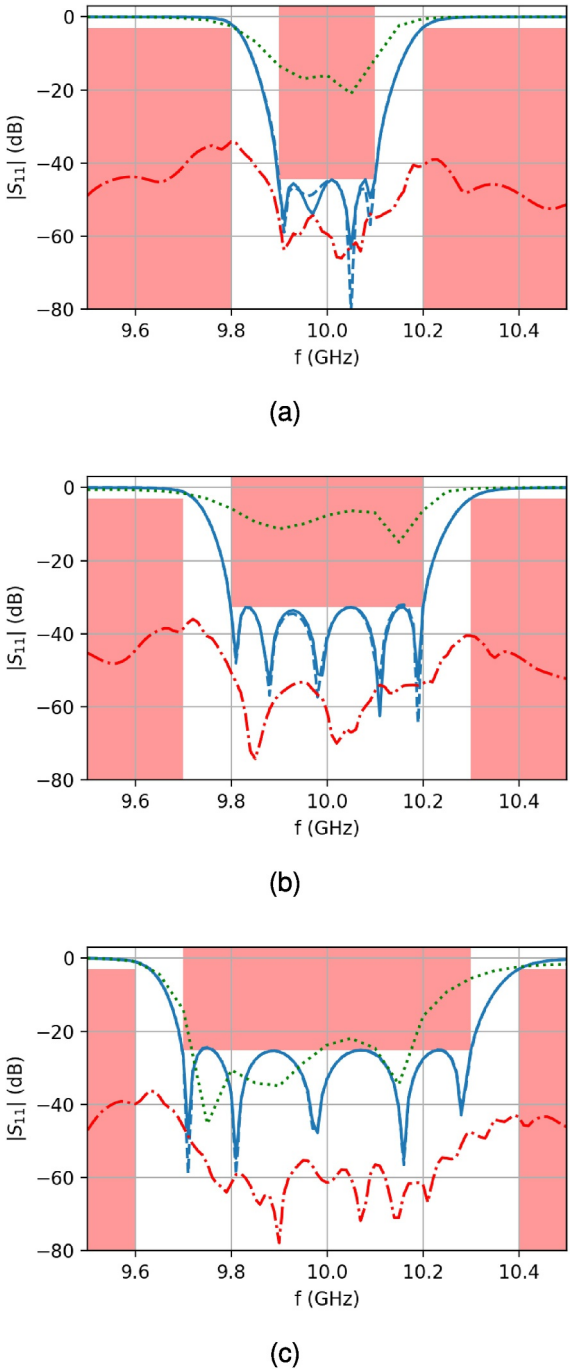

FIGURE 14 | Scattering parameter $|S_{11}|$ as a function of frequency for three different bandwidths (a) $B = 0.2$ GHz, (b) $B = 0.4$ GHz and (c) $B = 0.6$ GHz: (i) solid curve—$|\widehat{S}_{11}|$ as predicted by the GIANT network for the optimised geometry $\mathbf{p}_{\text{opt}}$, (ii) dashed curve—$|S_{11}|$ computed by the FEM for the optimised geometry $\mathbf{p}_{\text{opt}}$, (iii) dash-dotted curve—the error $|\widehat{S}_{11} - S_{11}|$ for the GIANT network when compared to the FEM and (iv) dotted curve—$|S_{11}|$ computed by the FEM for the best available geometry in the training set and the test set according to the optimisation problem (43). Here, the pass and stop bands are indicated by the red areas.

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 | Optimised geometry parameters for different bandwidths.

B (GHz)	w_1 (mm)	w_2 (mm)	w_3 (mm)	l_1 (mm)	l_2 (mm)	l_3 (mm)
0.2	5.121	7.250	7.806	15.333	17.772	18.131
0.4	4.926	6.906	7.391	14.865	17.270	17.638
0.6	4.777	6.613	7.045	14.487	16.809	17.181

The optimised geometry parameters that allow for geometrical perturbations of ± 10 μm are presented in Table 3. We note that the dimensions are similar to the corresponding result presented in Table 2 for the unperturbed case.

5 | Conclusion

We present the Gradient-Informed Attentive Normalisation Training (GIANT) framework that allows efficient training of very deep fully connected neural networks, which we use as expressive and fast surrogate models to efficiently replace full-wave simulation tools for microwave problems. As the core component of the GIANT framework, we introduce Attentive Normalisation (AttNorm), a dynamic reparameterisation procedure for the weight-bias parameter space of fully connected neural networks. AttNorm features normalisation layers that have the objective to maintain zero mean and a suitable standard deviation for the input data of each fully connected layer of the neural network. Here, the scaling of each normalisation layer is determined by means of a novel low-variance preserving scaling function that provides a more robust training procedure and is less sensitive to the choice of hyperparameters when compared to a conventional scaling function. As part of AttNorm, we also present a custom updating scheme that offers improved convergence during training. The updating scheme features three main steps that are executed in a loop: (i) an update of the parameters of the normalisation layers followed by a subsequent update of the weight-bias parameters of the fully connected layers such that the output of the neural network remains unchanged; (ii) an update of the internal momentum and velocity parameters of the Adam optimiser such that they are consistent with the reparameterised weight space and (iii) the application of the Adam optimiser [22] to optimise the weight matrices and bias vectors of the neural network with respect to a given loss function. In the context of surrogate models, very deep fully connected neural networks may model remarkably complex phenomena but a successful training is often limited by the amount of data that can be generated by the finite computational resources available. Here, we exploit Sobolev training with a loss function that involves the misfit of both function values and their first-order derivatives where the derivatives may be computed (essentially) for free with the aid of continuum sensitivity analysis when applied to for example reciprocal scattering problems.

We test the GIANT framework on two reciprocal microwave devices: (i) a six-port microwave cavity with a random medium that models a near-field microwave measurement device and (ii) an H-plane waveguide filter that is optimised under geometrical

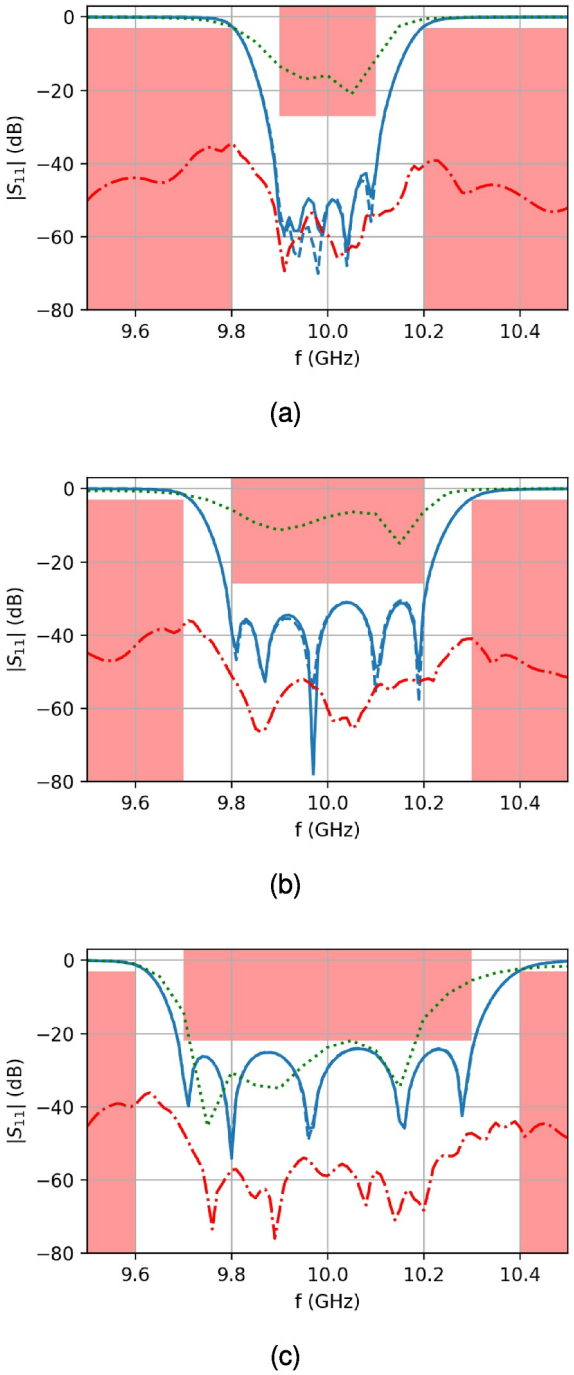

FIGURE 15 | Scattering parameter $|S_{11}|$ as a function of frequency for three different bandwidths (a) $B = 0.2$ GHz, (b) $B = 0.4$ GHz and (c) $B = 0.6$ GHz where the geometry parameters are subject to perturbations of ± 10 μm. Only curves for the unperturbed scattering parameters are shown: (i) solid curve—$|\widehat{S}_{11}|$ as predicted by the GIANT network for the optimised geometry $\mathbf{p}_{\text{opt}}$, (ii) dashed curve—$|S_{11}|$ computed by the FEM for the optimised geometry $\mathbf{p}_{\text{opt}}$, (iii) dash-dotted curve—the error $|\widehat{S}_{11} - S_{11}|$ for the GIANT network when compared to the FEM and (iv) dotted curve—$|S_{11}|$ computed by the FEM for the best available geometry in the training set and the test set according to the optimization problem (43). Here, the pass and stop bands are indicated by the red areas.

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 3 | Optimised geometry parameters that allow for geometrical perturbations for different bandwidths.

B (GHz)	w_1 (mm)	w_2 (mm)	w_3 (mm)	l_1 (mm)	l_2 (mm)	l_3 (mm)
0.2	5.135	7.256	7.803	15.366	17.789	18.140
0.4	4.919	6.896	7.382	14.879	17.277	17.645
0.6	4.775	6.604	7.036	14.507	16.814	17.185

uncertainty. Given a fixed computational budget, the GIANT framework enables faster and more accurate training of very deep fully connected neural networks (demonstrated up to 30 hidden layers) as compared to conventional training approaches available in the open literature. The trained neural networks achieve an error on an unseen test set that is comparable to the error of the field solver that is used to create the training, validation and test data sets. Thus, we conclude that the trained neural networks can be used as a satisfactory surrogate models for the field solver, making the GIANT framework an attractive alternative for a large group of problems that otherwise would be intractable to solve.

Author Contributions

Simon Stenmark: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing – original draft, writing – review and editing. **Thomas Rylander:** conceptualization, formal analysis, funding acquisition, methodology, project administration, resources, writing – original draft, writing – review and editing. **Tomas McKelvey:** conceptualization, methodology, writing – review and editing. **Andrei Ludvig-Osipov:** conceptualization, methodology, writing – review and editing.

Acknowledgements

The computations were enabled by resources provided by Chalmers e-Commons at Chalmers.

Funding

This work was supported by (i) the Swedish Governmental Agency for Innovation Systems (VINNOVA) under Grant 2016-00460 and (ii) Chalmers University of Technology.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are openly available in DORIS at https://doi.org/10.71870/8e3q-j519, reference number 2025-219.

References

1. A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method (3rd Edition)* (Artech House, 2005).

2. J. M. Jin, *The Finite Element Method in Electromagnetics (3rd Edition)* (John Wiley & Sons, 2014).

3. W. C. Chew, J. M. Jin, E. Michielssen, and J. M. Song, *Fast and Efficient Algorithms in Computational Electromagnetics* (Artech House, 2001).

4. S. Koziel and L. Leifsson, *Surrogate-Based Modeling and Optimization* (Springer, 2013).

5. Y. Yu, Z. Zhang, Q. S. Cheng, et al., "State-of-the-Art: AI-Assisted Surrogate Modeling and Optimization for Microwave Filters," *IEEE Transactions on Microwave Theory and Techniques* 70, no. 11 (November 2022): 4635–4651, https://doi.org/10.1109/tmtt.2022.3208898.

6. J. Bandler, R. Biernacki, S. H. Chen, P. Grobelny, and R. Hemmers, "Space Mapping Technique for Electromagnetic Optimization," *IEEE Transactions on Microwave Theory and Techniques* 42 (December 1994): 2536–2544, https://doi.org/10.1109/22.339794.

7. S. Koziel, J. Bandler, and Q. Cheng, "Constrained Parameter Extraction for Microwave Design Optimisation Using Implicit Space Mapping," *IET Microwaves, Antennas & Propagation* 5, no. 10 (July 2011): 1156–1163: Publisher: The Institution of Engineering and Technology, https://doi.org/10.1049/iet-map.2010.0607.

8. S. Koziel, A. Bekasiewicz, P. Kurgan, and J. W. Bandler, "Rapid Multi-Objective Design Optimisation of Compact Microwave Couplers by Means of Physics-Based Surrogates," *IET Microwaves, Antennas & Propagation* 10, no. 5 (April 2016): 479–486: Publisher: The Institution of Engineering and Technology, https://doi.org/10.1049/iet-map.2015.0279.

9. S. Koziel and A. Pietrenko-Dabrowska, "Rapid Surrogate-Aided Multicriterial Optimization of Compact Microwave Passives Employing Machine Learning and ANNs," *IEEE Transactions on Microwave Theory and Techniques* 72, no. 8 (August 2024): 4475–4488, https://doi.org/10.1109/tmtt.2024.3359703.

10. Q.-J. Zhang, K. Gupta, and V. Devabhaktuni, "Artificial Neural Networks for RF and Microwave Design—From Theory to Practice," *IEEE Transactions on Microwave Theory and Techniques* 51 (April 2003): 1339–1350, https://doi.org/10.1109/TMTT.2003.809179.

11. F. Feng, C. Zhang, J. Ma, and Q.-J. Zhang, "Parametric Modeling of EM Behavior of Microwave Components Using Combined Neural Networks and Pole-Residue-Based Transfer Functions," *IEEE Transactions on Microwave Theory and Techniques* 64, no. 1 (January 2016): 60–77, https://doi.org/10.1109/tmtt.2015.2504099.

12. Y. Chen, Y.-B. Tian, Z. Qiang, and L. Xu, "Optimisation of Reflection Coefficient of Microstrip Antennas Based on KBNN Exploiting GPR Model," *IET Microwaves, Antennas & Propagation* 12, no. 4 (March 2018): 602–606: Publisher: The Institution of Engineering and Technology, https://doi.org/10.1049/iet-map.2017.0282.

13. Z. Ye, W. Shao, X. Ding, B.-Z. Wang, and S. Sun, "Knowledge-Based Neural Network for Multiphysical Field Modeling," *IEEE Transactions on Microwave Theory and Techniques* 71, no. 5 (May 2023): 1967–1976, https://doi.org/10.1109/tmtt.2022.3227333.

14. K. S. Um, N. J. Kim, and S. W. Heo, "Surrogate-Based Model Using Auto-Encoder for Optimising Multi-Band Antennas," *IET Microwaves, Antennas & Propagation* 16, no. 11 (September 2022): 725–732: Publisher: The Institution of Engineering and Technology, https://doi.org/10.1049/mia2.12288.

15. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," preprint, arXiv:1502.03167 [cs] (March 2015).

16. H. Peng, Y. Yu, and S. Yu, "Re-Thinking the Effectiveness of Batch Normalization and Beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, no. 1 (January 2024): 465–478, https://doi.org/10.1109/tpami.2023.3319005.

17. E. S. Lubana, R. Dick, and H. Tanaka, "Beyond BatchNorm: Towards a Unified Understanding of Normalization in Deep Learning," in *Advances in Neural Information Processing Systems*, Vol. 34 (Curran Associates Inc., 2021), 4778–4791.

18. G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks," preprint, arXiv:1706.02515 [cs.LG] (September 2017).

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

19. G. Desjardins, K. Simonyan, R. Pascanu, and K. Kavukcuoglu, "Natural Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, Vol. 28 (2015), 2071–2079.

20. J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q.-J. Zhang, "Deep Neural Network Technique for High-Dimensional Microwave Modeling and Applications to Parameter Extraction of Microwave Filters," *IEEE Transactions on Microwave Theory and Techniques* 67, no. 10 (October 2019): 4140–4155, https://doi.org/10.1109/tmtt.2019.2932738.

21. E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa, "A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks," in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)* (December 2018), 92–99.

22. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," preprint, arXiv:1412.6980 [cs.LG] (January 2017).

23. F. Feng, Q. Guo, J. Chen, et al., "Feature and EM Sensitivity Co-Assisted Neuro-TF Surrogate Optimization for Microwave Filter Design," *IEEE Transactions on Microwave Theory and Techniques* 71, no. 11 (November 2023): 4749–4761, https://doi.org/10.1109/tmtt.2023.3275232.

24. J. Xu, M. Yagoub, R. Ding, and Q. J. Zhang, "Exact Adjoint Sensitivity Analysis for Neural-Based Microwave Modeling and Design," *IEEE Transactions on Microwave Theory and Techniques* 51 (January 2003): 226–237, https://doi.org/10.1109/TMTT.2002.806910.

25. S. A. Sadrossadat, Y. Cao, and Q.-J. Zhang, "Parametric Modeling of Microwave Passive Components Using Sensitivity-Analysis-Based Adjoint Neural-Network Technique," *IEEE Transactions on Microwave Theory and Techniques* 61, no. 5 (May 2013): 1733–1747, https://doi.org/10.1109/tmtt.2013.2253793.

26. W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Świrszcz, and R. Pascanu, "Sobolev Training for Neural Networks," preprint, arXiv:1706.04859 [cs.LG]. (July 2017).

27. A. Bondeson, Y. Yang, and P. Weinerfelt, "Shape Optimization for Radar Cross Sections by a Gradient Method," *International Journal for Numerical Methods in Engineering* 61, no. 5 (October 2004): 687–715, https://doi.org/10.1002/nme.1088.

28. Y. Yang, T. Halleröd, D. Ericsson, A. Hellervik, A. Bondeson, and T. Rylander, "Gradient Optimization of Microwave Devices Using Continuum Design Sensitivities From the Adjoint Problem," *IEEE Transactions on Magnetics* 41, no. 5 (May 2005): 1780–1783, https://doi.org/10.1109/tmag.2005.845993.

29. T. Rylander, P. Hashemzadeh, and M. Viberg, "Reconstruction of Metal Protrusion on Flat Ground Plane," *IET Microwaves, Antennas & Propagation* 4, no. 11 (2010): 1746–1755, https://doi.org/10.1049/iet-map.2009.0250.

30. X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (May 2010), 249–256.

31. M. Abadi, A. Agarwal, P. Barham, et al., "TensorFlow," accessed November 21, 2025, https://www.tensorflow.org.

32. J. Schöberl, M. Hochsteger, and C. Lackner, "NGSolve," accessed November 21, 2025, https://www.ngsolve.org.

33. J. Nohlert, T. Rylander, and T. McKelvey, "Microwave Measurement System for Detection of Dielectric Objects in Powders," *IEEE Transactions on Microwave Theory and Techniques* 64, no. 11 (2016): 3851–3863, https://doi.org/10.1109/tmtt.2016.2613047.

34. J. Winges, L. Cerullo, T. Rylander, T. McKelvey, and M. Viberg, "Compressed Sensing for the Detection and Positioning of Dielectric Objects Inside Metal Enclosures by Means of Microwave Measurements," *IEEE Transactions on Microwave Theory and Techniques* 66, no. 1 (2017): 462–476, https://doi.org/10.1109/tmtt.2017.2708109.

17518733, 2026, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/mia2.70077 by Chalmers Research - <shibboleth>-member@chalmers.se , Wiley Online Library on [16/03/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License