Electromagnetic-Thermal Analysis With FDTD and Physics-Informed Neural Networks

Shutong Qi, *Graduate Student Member, IEEE*, and Costas D. Sarris, *Senior Member, IEEE*

***Abstract*—This article presents the coupling of the finite-difference time-domain (FDTD) method for electromagnetic field simulation, with a physics-informed neural network based solver for the heat equation. To this end, we employ a physics-informed U-Net instead of a numerical method to solve the heat equation. This approach enables the solution of general multiphysics problems with a single-physics numerical solver coupled with a neural network, overcoming the questions of accuracy and efficiency that are associated with interfacing multiphysics equations. By embedding the heat equation and its boundary conditions in the U-Net, we implement an unsupervised training methodology, which does not require the generation of ground-truth data. We test the proposed method with general 2-D coupled electromagnetic-thermal problems, demonstrating its accuracy and efficiency compared to standard finite-difference based alternatives.**

***Index Terms*—Neural networks, unsupervised learning, FDTD, electrothermal problems.**

I. Introduction

MULTIPHYSICS interactions of electromagnetic fields are important in a wide range of applications, from biochemical interactions of microwaves [1] to microwave ablation [2]. In such cases, electromagnetic field computations are combined with computations of associated thermal and biological effects. Coupled electromagnetic-thermal problems can be analyzed by numerical solvers, such as finite difference or finite element methods [3], [4], [5] for Maxwell's equations and the heat equation. The stability and accuracy of techniques such as the Finite-Difference Time-Domain (FDTD) method and explicit/implicit methods for the heat equation are well understood in the context of single-physics simulations. However, error propagation between coupled multiphysics numerical techniques has received limited attention in the literature [6]. Moreover, the spatial discretization that is typically used is inherently sub-optimal, as the standard approach is to generate the mesh for the electromagnetic solver and reuse it in the thermal solver.

Scientific machine learning is a rapidly expanding area, where artificial intelligence (AI) algorithms are employed to solve problems in computational science and engineering. Major breakthroughs have been enabled by harnessing massive data from measurements and simulations [7], [8], [9]. Within this area, physics-informed neural networks (PINNs) are aimed at the solution of ordinary/partial differential equations and inverse problems [10], [11], without invoking conventional numerical methods.

In this paper, we explore the use of PINNs as a component of a multiphysics solver for electromagnetic-thermal problems. In particular, we use the FDTD method for electromagnetic field simulation, but we use a PINN as a thermal solver. This approach combines the robustness of an established and well-understood numerical technique for Maxwell's equations (FDTD) with the computational efficiency of a PINN that is trained to rapidly and accurately solve the heat equation. Moreover, we enable the solution of electromagnetic-thermal problems by standalone electromagnetic simulators, as the heat equation solver is replaced by the PINN.

In particular, we train two U-Nets [12] in an unsupervised manner. The first U-Net determines the temperature as a function of time, for applications such as microwave ablation that require the computation of the time needed to achieve a certain temperature level within a tissue. The second U-Net directly determines the steady-state temperature profile for applications such as the design optimization of microwave ablation antennas to engineer a desired steady-state isotherm pattern within a tissue [2]. Both problems are solved within an inhomogeneous, lossy medium that is specified as an input to each U-Net. The FDTD method is employed to compute the source term for the heat equation, namely the dissipated power density on a time-averaged basis, which is also included as an input to the U-Nets. Unlike [13], [14], we use a totally unsupervised strategy, where the heat equation is applied as the loss function and the boundary conditions are implemented into the U-Net architecture. Hence, the training process does not require the generation of ground-truth data. Fig. 1 shows the flowchart of the proposed methodology.

The rest of this paper is organized as follows. In Section II, we briefly review FDTD-based electromagnetic-thermal simulations and the structure of convolutional neural networks. In Section III, the architecture of the proposed physics-informed U-Net is shown, along with the implementation of the heat equation and boundary conditions into the U-Net. In Section IV, we demonstrate the accuracy and feasibility of our proposed methodology via numerical experiments. Its computational efficiency and generalizability with respect to geometry and material properties of the simulated media are discussed in Section V. We summarize our results and contributions in Section VI.

Manuscript received 27 November 2022; revised 18 December 2022; accepted 9 January 2023. Date of publication 13 January 2023; date of current version 1 February 2023. This work was supported in part by the National Sciences and Engineering Research Council of Canada (NSERC) under the Discovery Grand RGPIN-2018-04650. *(Corresponding author: Shutong Qi.)*

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: st.qi@mail.utoronto.ca; costas.sarris@utoronto.ca).

Digital Object Identifier 10.1109/JMMCT.2023.3236946

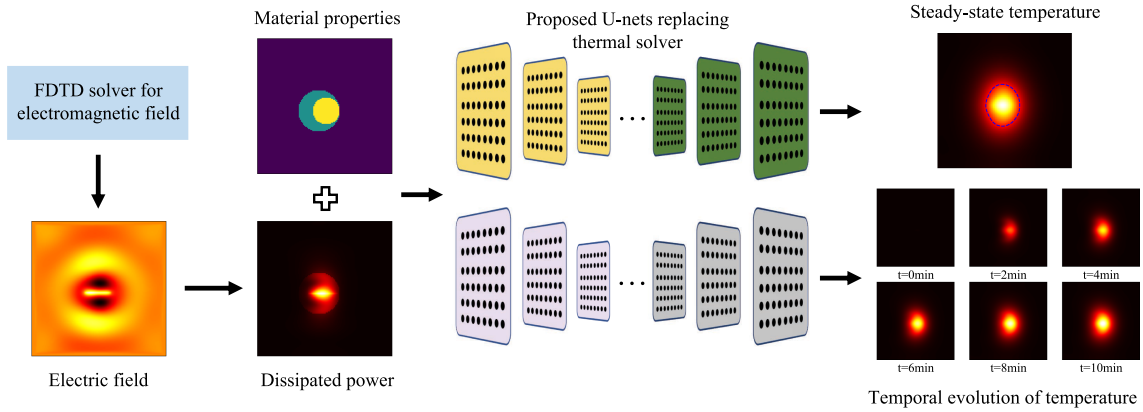


Fig. 1. The flowchart of the proposed method.

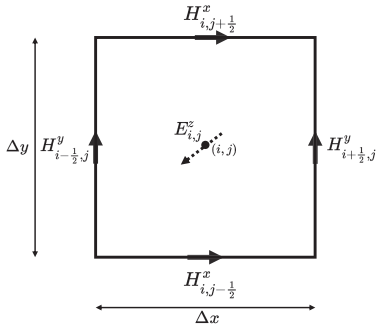


Fig. 2. 2D Yee cell.

II. Background and Overview of This Work

In this section, we give an overview of the 2-D FDTD update equations for lossy media and a review of the structure of convolutional neural networks and U-Nets.

A. Review of Finite Difference Based Electromagnetic-Thermal Simulations

The finite-difference time-domain (FDTD) method is widely used in solving electromagnetic-thermal problems [3], [4]. In the following, we briefly review the use of FDTD and the forward-time centered-space (FTCS) method for coupled electromagnetic-thermal problems. Without loss of generality, we focus on transverse magnetic (TM) polarization.

Based on the Yee's cell of Fig. 2, we use centered finite differences to approximate the temporal and spatial derivatives of Maxwell's curl equations. Then, the update equation for the $z-$component of the electric field, E^z in terms of the magnetic field components H^x, H^y, is:

$$E^{z,n+1}_{i,j} = \frac{\frac{\varepsilon_{i,j}}{\Delta t} - \frac{\sigma_{i,j}}{2}}{\frac{\varepsilon_{i,j}}{\Delta t} + \frac{\sigma_{i,j}}{2}} E^{z,n}_{i,j} + \frac{1}{\frac{\varepsilon_{i,j}}{\Delta t} + \frac{\sigma_{i,j}}{2}} \cdot \left(\frac{H^{y,n+\frac{1}{2}}_{i+\frac{1}{2},j} - H^{y,n+\frac{1}{2}}_{i-\frac{1}{2},j}}{\Delta x} - \frac{H^{x,n+\frac{1}{2}}_{i,j+\frac{1}{2}} - H^{x,n+\frac{1}{2}}_{i,j-\frac{1}{2}}}{\Delta y} \right), \tag{1}$$

where ε is the permittivity, σ is the electrical conductivity, Δt is the time step, Δx and Δy are the dimensions of the Yee cell. The update equation also includes the spatial indices (i, j) and the time step index n.

For the magnetic field, the update equations are:

$$H^{x,n+\frac{1}{2}}_{i,j+\frac{1}{2}} = H^{x,n-\frac{1}{2}}_{i,j+\frac{1}{2}} + \frac{\Delta t}{\mu} \frac{E^{z,n}_{i,j+1} - E^{z,n}_{i,j}}{\Delta y}, \tag{2}$$

$$H^{y,n+\frac{1}{2}}_{i+\frac{1}{2},j} = H^{y,n-\frac{1}{2}}_{i+\frac{1}{2},j} - \frac{\Delta t}{\mu} \frac{E^{z,n}_{i+1,j} - E^{z,n}_{i,j}}{\Delta x}, \tag{3}$$

where μ is the permeability.

The stability condition is:

$$\Delta t \leq \frac{1}{v_{\max}} \frac{1}{\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}}, \tag{4}$$

where $v_{\max}$ is the maximum wave phase velocity in the simulation domain.

Next, the FTCS-based solution of the heat equation is reviewed. In the presence of a cooling fluid circulating inside a medium, the heat equation can be written as:

$$\rho C_p \frac{\partial T}{\partial t} = Q + \nabla \cdot (k \nabla T) + V_s (T_b - T) \tag{5}$$

where ρ and C_p represent the material density and heat capacity, respectively. T is the temperature, Q is the power density of the heat source and k is the heat conductivity. V_s is the product of flow and heat capacity of the cooling fluid and T_b is the temperature of the cooling fluid. In this problem, the heat source comes from the electromagnetic field, where $Q = \sigma \left\langle |\mathbf{E}|^2 \right\rangle$, and $\langle \cdot \rangle$ denotes time average.

As mentioned above, the standard approach, also adopted here, is to reuse the FDTD discretization of the working volume to solve the heat equation [3], [4]. By applying the FTCS method, the update equation for the temperature is:

$$T^{n+1}_{i,j} = T^n_{i,j} + \frac{k \Delta t}{\rho C_p} \cdot \left[\frac{T^n_{i+1,j} - 2T^n_{i,j} + T^n_{i-1,j}}{\Delta x^2} + \frac{T^n_{i,j+1} - 2T^n_{i,j} + T^n_{i,j-1}}{\Delta y^2} + \frac{Q_{i,j} + V_s (T_b - T^n_{i,j})}{k} \right], \tag{6}$$

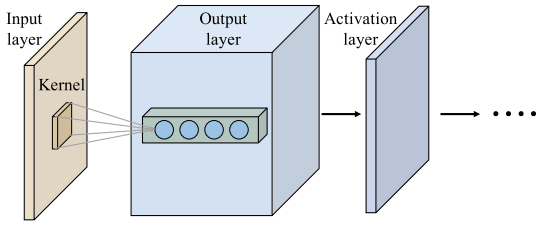


Fig. 3. The architecture of a convolutional layer.

and

$$Q_{i,j} = \sigma_{i,j} \left\langle |\mathbf{E}_{i,j}|^2 \right\rangle . \tag{7}$$

The stability condition for (6) is:

$$\Delta t \leq \frac{1}{2D_t} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1},$$
$$D_t = \frac{k}{\rho C_p}. \tag{8}$$

We assume that all materials included in the following simulations are temperature-independent. When the electromagnetic field reaches steady state, the dissipated power density ($Q_{i,j}$) is determined throughout the computational domain and used as a source in the heat equation.

B. U-Net and Convolutional Neural Network

U-Net is an encoder-decoder structure with a down-sampling route and an up-sampling route. In each route, there are several blocks consisting of convolutional layers. A convolutional layer is formulated by locally weighted summations of the input data, as shown in Fig. 3. For a single weighted calculation, the inner product between one local batch and a convolution filter is performed, followed by a nonlinear activation function [15], [16]. We set $X^{(l-1)}(m \times m)$ as the input of the $(l-1)$-th layer, $K^{(l)}(n \times n)$ as the corresponding filter kernel, and b as the bias of the kernel. The output of this layer, $Z^{(l)}((m-n+1) \times (m-n+1))$, consists of elements:

$$Z^{(l)}_{u,v} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} X^{(l-1)}_{i+u,j+v} \cdot K^{(l)}_{i,j} \cdot \chi(i,j) + b^{(l)} \tag{9}$$

$$\chi(i,j) = \begin{cases} 1, 0 \leq i,j \leq n \\ 0, \text{otherwise.} \end{cases} \tag{10}$$

After a convolution operation, the output needs to be processed by the activation layer:

$$a^{(l)}_{u,v} = g\left(z^{(l)}_{u,v}\right), \tag{11}$$

where $g(x)$ is a nonlinear activation function.

Generally, the convolution filter is smaller than the size of the feature map. It incorporates a matrix of trainable weights, which are shared by two locally connected layers, allowing the network to include more layers without the burden to store a large number of parameters. By stacking many convolutional layers together, we can extract multi-level information. With these features, the convolutional neural network is very efficient in processing data structures with large size and multiple dimensions, such as two-dimensional and three-dimensional images.

In the U-Net, down-sampling is enabled by the max-pooling layer between each block in the down-sampling route. A max-pooling layer takes the maximum value over an input window. Moreover, there are transposed convolutional layers between each block in the up-sampling route. The process of a transposed convolutional layer is opposite to a convolutional layer. It can broadcast input elements via a kernel, thereby reconstructing the output from convolutional layers. The U-Net structure used in this work will be discussed in detail, in Section III.

III. U-Net as a Thermal Solver Coupled to FDTD

We replace the thermal solver with two U-Nets in the electromagnetic-thermal simulation. The first U-Net predicts the temporal evolution of temperature within the computational domain and the second U-Net can directly generate the steady-state temperature distribution. The architecture of these two U-Nets is identical except for the number of channels of the input and output data, which will be discussed next.

The highlight of the proposed method is the unsupervised training of the U-Nets. We use the transformed heat equation as a loss function and implement the boundary conditions of the temperature into the proposed U-Nets. Then, the network parameters can be optimized through backpropagation based on the discrete heat equation instead of ground-truth data like the method proposed in [17]. Next, we will discuss how to implement the heat equation and boundary conditions into the U-Nets and will show the structure of the proposed U-Nets.

A. Implementation of the Heat Equation as a Loss Function

The temporal evolution of temperature is governed by the heat (5), and can be numerically determined by the FTCS method (6). From (8), we know that the time step of FTCS is limited by the stability condition. To accelerate the simulation, we train a U-Net to predict N time steps of temperature evolution simultaneously. Since this U-Net predicts the evolution of temperature in the time domain, we call it time-domain U-Net (TDUN). The comparison between standard FTCS and the proposed TDUN is shown in Fig. 4.

We use a modified heat equation to replace the normal mean square error when training the TDUN, so that the U-Net can be trained in an unsupervised manner. To be specific, if we use a forward finite difference to approximate the partial derivative of the temperature with respect to time in (5), the update equation for temperature is:

$$T^n - T^{n+1} + \frac{\Delta t_{div}}{\rho C_p} \left(Q + k\nabla^2 T^n + V_s \left(T_b - T^n \right) \right) = 0. \tag{12}$$

All the variables in (12) are the same as in (5). The left-hand side of (12) can be used to optimize the predicted temperature for one time step. For N successive steps of temperature update,

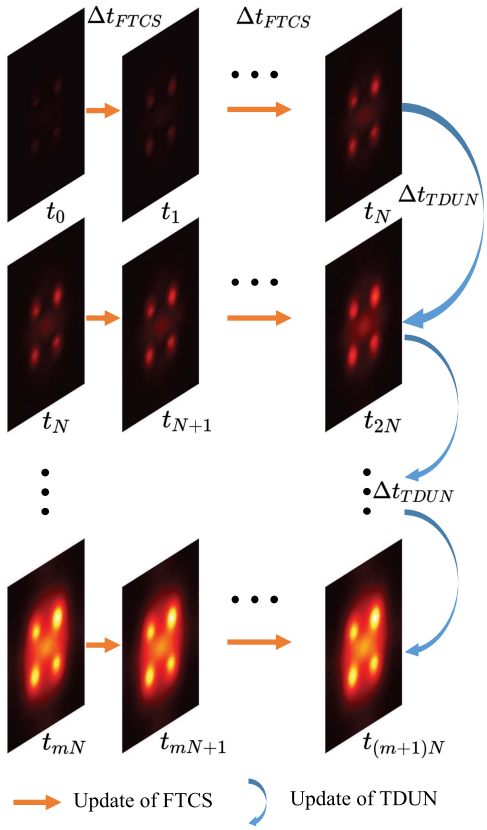


Fig. 4. Comparison between FTCS and the proposed TDUN.

the summation of all steps is:

$$\sum_{i=n}^{n+N-1}\left(T^i - T^{i+1} + \frac{\Delta t_{div}}{\rho C_p}(Q + (k\nabla^2 T^i) + V_s(T_b - T^i))\right) = 0. \tag{13}$$

We use (13) to replace the normal mean square error in the TDUN. Then, the loss function for this network is:

$$L_{TDUN} = \sum_{i=n}^{n+N-1}\left|T^i - T^{i+1} + \frac{\Delta t_{div}}{\rho C_p}(Q + (k\nabla^2 T^i) + V_s(T_b - T^i))\right|^2. \tag{14}$$

L_{TDUN} tends to 0, as the predicted temperature approaches the actual solution. Otherwise, the TDUN will tune the parameters through backpropagation to decrease (14). Fig. 5 shows how this procedure works.

The batch size N in the TDUN is a hyperparameter set by the user, where $\Delta t_{TDUN} = N\Delta t_{div}$. Evidently, N and Δt_{div} affect the accuracy and efficiency of the TDUN. We will study these two hyperparameters in Section IV, where we will also show that the time step Δt_{div} is not restricted by the FTCS stability limit (8).

Since the TDUN simulates the dynamic evolution of temperature in the computational domain, we can compute the temperature distribution at each time step and the heating time

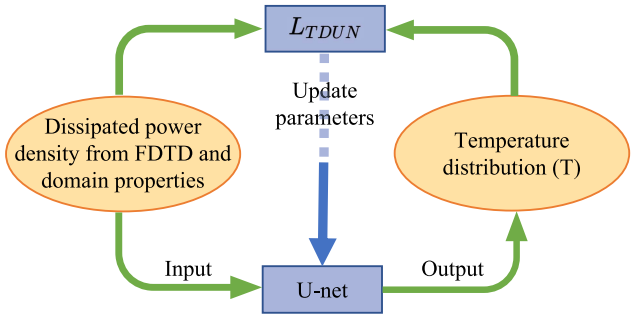


Fig. 5. Implementation of the heat equation as a loss function.

that is needed for a temperature profile to evolve. However, if we are only interested in the steady-state temperature distribution within the computational domain, we can follow an alternative, more efficient approach. To this end, we propose a second U-Net based neural network to directly predict the steady-state temperature distribution within the computational domain. Since this network predicts steady-state temperature distribution, we call it steady-state U-Net (SSUN).

The SSUN is also trained in an unsupervised manner. At steady state, the heat (5) becomes:

$$Q + k\nabla^2 T + V_s(T_b - T) = 0. \tag{15}$$

We use (15) as the loss function in the SSUN to enable its unsupervised training. Then, the loss function for the SSUN is:

$$L_{SSUN} = \left|Q + k\nabla^2 T + V_s(T_b - T)\right|^2. \tag{16}$$

We implement the heat equation as the loss function for both the heat diffusion and the steady-state simulation. Next, the implementation of boundary conditions will be discussed.

B. Implementation of Boundary Conditions Into a Neural Network

Boundary conditions can be directly embedded into the architecture of the U-Nets. For thermal problems, there are three kinds of boundary conditions: Dirichlet (known boundary temperature), Neumann (known boundary heat flux) and Robin (known heat transfer coefficient and external temperature). It is easy to implement the Dirichlet boundary condition by assigning a fixed temperature to the boundary of the computational domain. The Neumann boundary condition is a special case of the Robin boundary condition, so only the implementation of the latter will be discussed in the following.

For the Robin boundary condition,

$$-k\left(\frac{\partial T}{\partial \mathbf{n}}\right)_w = h\left(T_w - T_f\right) \tag{17}$$

where $\partial T/\partial \mathbf{n}$ is the partial derivative of T in the normal direction defined by $\mathbf{n}$, w represents the position of the boundary, h is the convective heat transfer coefficient, T_w is the boundary temperature and T_f is the external temperature. In the x-direction,

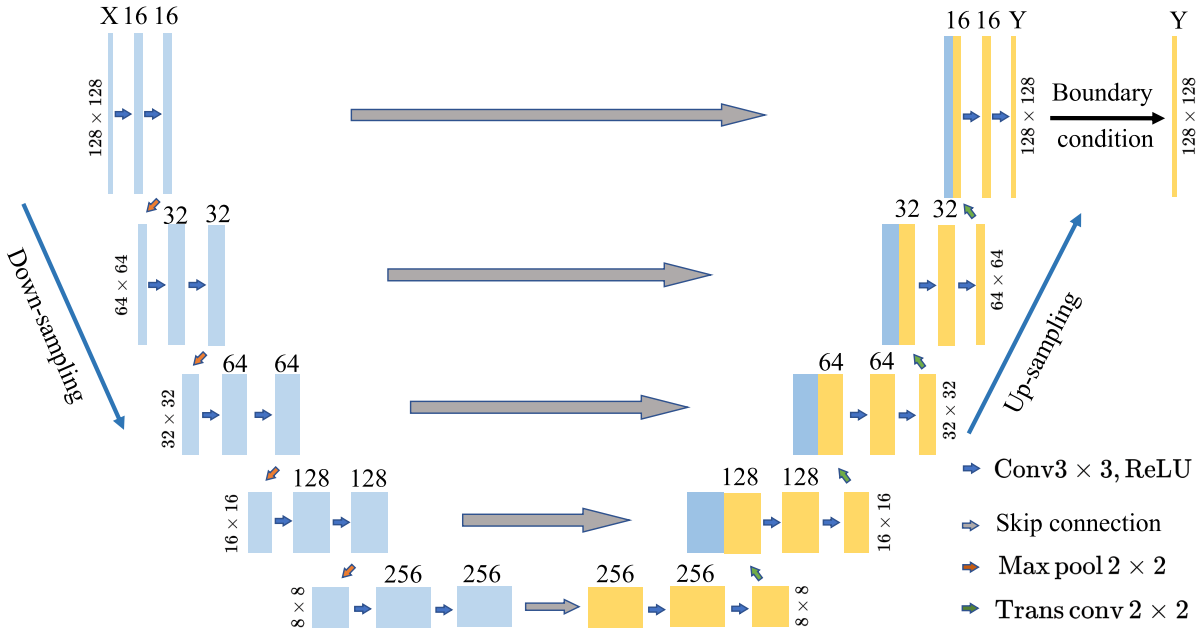


Fig. 6. Detailed architecture of the proposed time-domain U-Net (TDUN) and steady-state U-Net (SSUN). X = 5, Y = N for TDUN and X = 2, Y = 1 for SSUN.

(17) can be discretized as:

$$-k\left(\frac{T_{i,j}-T_{i-1,j}}{\Delta x}\right)=h\left(\frac{T_{i,j}+T_{i-1,j}}{2}-T_f\right), \tag{18}$$

where $T_{i,j}$ represents the temperature at the boundary and $T_{i-1,j}$ is the temperature on the left of $T_{i,j}$. Then, the temperature at (i, j) is

$$T_{i,j}=\frac{(2k-h\Delta x)T_{i-1,j}+2h\Delta x T_f}{2k+h\Delta x}. \tag{19}$$

These expressions can be directly embedded into the U-Nets. Next, we will discuss the architecture of the proposed U-Nets.

C. Proposed Physics-Informed Neural Network Structure

The proposed TDUN and SSUN are built based on the U-Net architecture. They have identical architecture except for the number of channels of input and output data. U-Net is broadly used in scientific machine learning based field prediction [18], [19], [20]. The detailed structure of the proposed U-Nets is shown in Fig. 6.

The input of the TDUN is a 5-layer matrix, which represents initial temperature distribution $T^0\,(\mathrm{K})$, dissipated power density $Q\,(\mathrm{W/m^3})$, material density $\rho\,(\mathrm{kg/m^3})$, heat conductivity $k\,(\mathrm{W/m/K})$, and heat capacity $C_p\,(\mathrm{J/Kg/K})$ of the computational domain, respectively. The output of the TDUN is an N-layer matrix, which contains N successive steps of temperature $\{T_{i,j}\}^{n=1\ldots N}\,(\mathrm{K})$. Therefore, X = 5 and Y = N for the TDUN in Fig. 6.

From (15), it is observed that the material density and heat capacity are not involved in the steady-state heat equation. Therefore, the input of the SSUN is a 2-layer matrix, which contains the dissipated power density $Q\,(\mathrm{W/m^3})$ and the heat conductivity $k\,(\mathrm{W/m/K})$ of the computational domain. The output of the SSUN is the steady-state temperature distribution of the computational domain. So, X = 2 and Y = 1 for the SSUN in Fig. 6.

A U-Net has two branches, a down-sampling route and an up-sampling route. In the down-sampling branch, the U-Net extracts the information of the dissipated power density in the domain and material properties from the input data and transforms them into multi-channel features with a higher dimensional representation. The up-sampling branch learns from these features and reconstructs the multi-channel data back to the temperature distribution.

There are 5 corresponding blocks in the two branches. In each block, there are two 3 × 3 convolution filters with a ReLU activation function. The padding size of these two convolutional layers is 1 on each boundary, and the step size of convolutional kernels is also 1. So, the output dimension does not change after each block. In the down-sampling route, there is a max-pooling layer between two neighboring blocks, which can decrease the size of feature maps. The kernel size of the max-pooling layer is 2 × 2. Therefore, after each down-sampling block, the feature map is halved in each dimension. In the up-sampling route, there is a transposed convolutional layer between each block. The kernel size of this layer is also 2 × 2. So, after each block, the size of the feature map is doubled in each dimension.

Finally, between the down-sampling route and the up-sampling route, there is a skip connection between the corresponding block. In Fig. 6, we can see that the skip connection copies the output from one down-sampling route to the corresponding block in the up-sampling route directly. This direct connection will keep different level features in the training process.

TABLE I
ELECTRICAL PROPERTIES OF MATERIALS IN THE SIMULATION AT 2 GHZ

	Relative Permittivity ε_r	Relative Permeability μ_r	Electrical Conductivity σ(S/m)
Breast tissue	8.163 ± 0.577	1	0.497 ± 0.069
Benign Tumor	21.664 ± 1.559	1	0.955 ± 0.122
Cancer	63.008 ± 2.108	1	4.164 ± 0.074

TABLE II
THERMAL PROPERTIES OF MATERIALS IN THE SIMULATION

	Heat Conductivity $k\,(\mathrm{W/m/K})$	Heat Capacity $C_p\,(\mathrm{J/Kg/K})$	Density $\rho\,(\mathrm{kg/m^3})$
Breast	0.500	3770	1000±100
Benign Tumor	0.580±0.02	3600±200	1000±100
Cancer	0.580±0.02	3600±200	1000±100
Blood		3622.5	1000±100

After applying the boundary conditions to the U-Net outputs, we get the predicted temperature distribution. The loss is computed by (14) and (16). The Adam optimizer [21] is used for backpropagation to minimize the loss of the TDUN and SSUN.

IV. NUMERICAL RESULTS

In this section, we demonstrate the accuracy of the proposed TDUN and SSUN through two-dimensional cases: simulating the temperature distribution of a lossy object within a host medium stimulated by an electromagnetic wave. The material properties of the media in the computational domain correspond to those of breast tissue (for the host medium) and breast tumor (for the object). Therefore, the numerical experiments can be seen as a two-dimensional case study of microwave ablation of breast tumor slices. For this application, the heat equation in (5) is modified to the Pennes Bioheat Equation [22]:

$$\rho C_p \frac{\partial T}{\partial t} = Q + \nabla \cdot (k \nabla T) + W_b C_b (T_b - T), \tag{20}$$

where W_b, C_b, and T_b are the perfusion rate, heat capacity, and temperature of the blood. The term $W_b C_b (T_b - T)$ accounts for the cooling effect of blood perfusion through tissue.

In our numerical experiments, the operation frequency of the electromagnetic wave is 2 GHz. Tables I and II show the electromagnetic and thermal properties of the materials used in the simulations at the operation frequency respectively, which are taken from [23] and [24]. Besides, the initial temperature of the computational domain $T^0_{i,j}$ is set to 310 K. The perfusion rate of blood W_b is 1.75 $\mathrm{kg/s/m^3}$ and the blood temperature T_b is 310 K. The Robin boundary condition (19) is applied to the output temperature in TDUN and SSUN and the heat transfer coefficient h is 5 $\mathrm{W/m^2/K}$ according to [25].

Because the electromagnetic properties of benign and malignant tumors are different, the electromagnetic source amplitude has different settings for a desired steady-state temperature profile. According to [26], the induced temperature should be in excess of 323.15 K (50 °C) to ablate tumor cells. Based on a prior test, the source amplitude is set to 300 V/m for benign tumors and 100 V/m for malignant tumors. In the following, we will introduce the format of the input and output data as well as the results of the proposed U-Nets.

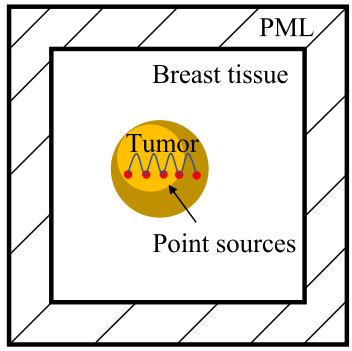


Fig. 7. The computational domain for the numerical example.

A. Data Format

The FDTD computational domain for the electromagnetic simulation is a square that is partitioned into 148 × 148 cells of dimensions $\Delta x = \Delta y = 0.9369$ mm, as shown in Fig. 7. The surrounding 10 cells belong to a perfectly matched layer (PML) [27]. In the computational region, we simulate the electromagnetic field in the lossy medium, stimulated by point sources. The object is represented by a single circle, ellipse or several overlapping circles and ellipses in the training set. This multitude of options allows us to generate a significant number of training cases for the U-Nets. In the testing set, we introduce multiple independent objects in the computational domain. The domain without the PML is divided into four 64 × 64 sub-domains uniformly. In each sub-domain, the object exists with a 50% probability.

The sources are placed within the object simulating a microwave ablation scenario [26], [28]. The number of point sources depends on the size of the object. For circular objects, we place the point sources along a segment of length equal to the radius of the circle, centered at the center of the circle. For ellipses, the sources are placed along the main axis from the center to half the semi-major axis on each side. If the object is composed of several overlapping circles or ellipses, we choose the largest object to set the sources. The region of the heat diffusion simulation is the same as the electromagnetic region without the PML. The boundary condition for temperature is the Robin boundary condition (17).

Each input of the TDUN is represented by a 5-layer matrix. The first 2 layers contain the initial temperature distribution $T^0\,(\mathrm{K})$ and the dissipated power density $Q\,(\mathrm{W/m^3})$ of the computational domain. The remaining 3 layers have the entries of material density $\rho\,(\mathrm{kg/m^3})$, heat capacity $C_p\,(\mathrm{J/Kg/K})$ and heat conductivity $k\,(\mathrm{W/m/K})$ of the computational domain, respectively. The input of the SSUN is a 2-layer matrix, which has the dissipated power density $Q\,(\mathrm{W/m^3})$ and heat conductivity $k\,(\mathrm{W/m/K})$ in the corresponding layer. From Tables I

and II, we can see that some material properties are uncertain. We consider these values as random variables that are uniformly distributed within their given range. If we take the example of heat conductivity k (W/m/K), $k_{i,j} = 0.58 + \zeta 0.02$ for the object and $k_{i,j} = 0.50$ for the host medium, where ζ is a uniformly distributed random variable from -1 to 1. For inhomogeneous cases, different random numbers are applied to regions with different properties.

The dimensions of each layer are 128 × 128. The dissipated power density is computed by FDTD. Since the input dimension of the U-Net is the same as that of the FDTD output without the PML, the dissipated power density from FDTD can be directly used in the U-Nets. The outputs of the U-Nets are temperature distribution in the computational domain, represented by a single/multiple-layer matrix with dimensions of 128 × 128 in each layer. The outputs of the TDUN present how the temperature evolves in the next N steps following the initial state. The output of the SSUN is the steady-state temperature distribution in the computational domain.

To simplify the training process, we use $T' = T - T_b$ to replace the original T. Then, the Pennes Bioheat (20) becomes:

$$\rho C_p \frac{\partial T'}{\partial t} = Q + k\nabla^2 T' - W_b C_b T'. \tag{21}$$

In this way, we avoid the influence of T_b in the data normalization process. Data normalization [29] is a common but important operation for deep neural networks. All the inputs and outputs of the network should be on a similar scale to facilitate the training process.

For the TDUN, we normalize the input dissipated power density by $Q' = Q/(\rho C_p)$. Besides, the material density ρ and heat capacity C_p are normalized by dividing by the largest number in the corresponding layer. The normalized density and heat capacity are denoted by ρ' and C_p', respectively. The scale factors are:

$$\begin{aligned} S_\rho &= \max\{\rho_{i,j}\}, \\ S_C &= \max\{C_{p\,i,j}\}. \end{aligned} \tag{22}$$

In this way, the input Q', ρ', C_p' and k have similar scales. The loss function of TDUN (14) is updated to:

$$\begin{aligned} L_{TDUN} = \sum_{i=n}^{n+N-1} \Big| & T'^{\,i} - T'^{\,i+1} + \Delta t_{div} Q' \\ & + \frac{\Delta t_{div}}{S_\rho \rho' S_C C_p'} \left((k\nabla^2 T'^{\,i}) - W_b C_b T'^{\,i} \right) \Big|^2. \end{aligned} \tag{23}$$

For the SSUN, we normalize the input power density by the largest number in the domain. Then, the normalized power density $Q'_{i,j} = Q_{i,j}/\max\{Q_{i,j}\}$, and the scale factor $S = \max\{Q_{i,j}\}$. With this normalization, all input parameters are in the range [0,1]. Then, the loss function of the SSUN (16) is updated to:

$$L_{SSUN} = \left| SQ' + k\nabla^{2\sim} T' - W_b C_b T' \right|^2. \tag{24}$$

(23) and (24) are the modified loss functions that are used to train the TDUN and SSUN.

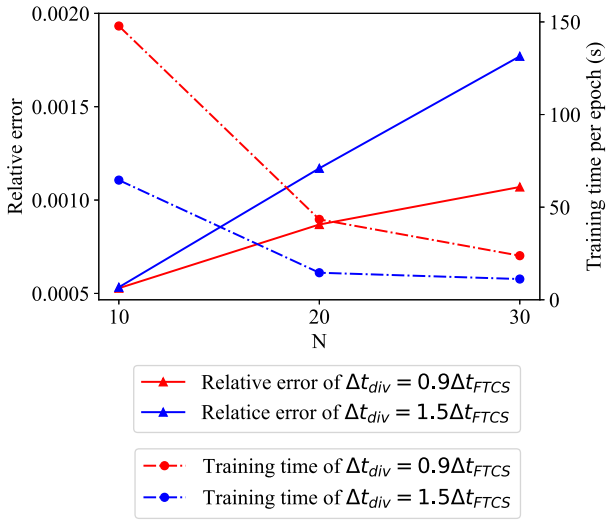


Fig. 8. Relative error and training time with respect to N and Δt_{div}.

B. Numerical Results

We build a dataset containing 1000 sets of training data and 200 sets of testing data. Training data include single homogeneous or inhomogeneous objects, while testing data have additional multiple objects in the computational region. So, the testing geometries are generalized versions of the training ones, with enhanced complexity. To evaluate the accuracy of the prediction, we use the FTCS-based thermal solver (6) to generate ground-truth data for the testing cases. Note that these are generated for validation, *not for training*, purposes, since our training is unsupervised. The relative error ($\mathcal{E}_{rel}$) is computed by:

$$\mathcal{E}_{rel} = \frac{1}{128^2} \sum_{i=1}^{128} \sum_{j=1}^{128} \frac{|T_p(i,j) - T_r(i,j)|}{T_r(i,j) - 310}, \tag{25}$$

where T_p is the prediction from the U-Nets and T_r is the result from FDTD.

When we train the TDUN, it is observed that the time step Δt_{div} in each batch is not limited by the stability condition in (8). Besides, the relative error and training time of the TDUN depend on both Δt_{div} and the batch size N. To explore how the relative error and training time vary with different N and Δt_{div}, we train the TDUN with $N = 10, 20, 30$ and $\Delta t_{div} = 0.9\Delta t_{FTCS}, 1.5\Delta t_{FTCS}$, where Δt_{FTCS} is the largest time step allowed by the stability condition (8). The training time records the time for training the TDUN with 240 sets of data per epoch. The runtime of the FDTD solver is not included in the reported training time, as we simulate the dissipated power density with FDTD once for each case, using the results in any training epoch. The relative error is the mean square error of each step on the testing set. The results are shown in Fig. 8. Fig. 8 shows that as we increase N, the relative error increases and the training time decreases regardless of Δt_{div}. The error growth rate of $\Delta t_{div} = 1.5\Delta t_{FTCS}$ is larger than that of $\Delta t_{div} = 0.9\Delta t_{FTCS}$. Moreover, as Δt_{div} increases, the relative error increases and the training time decreases. Therefore, the

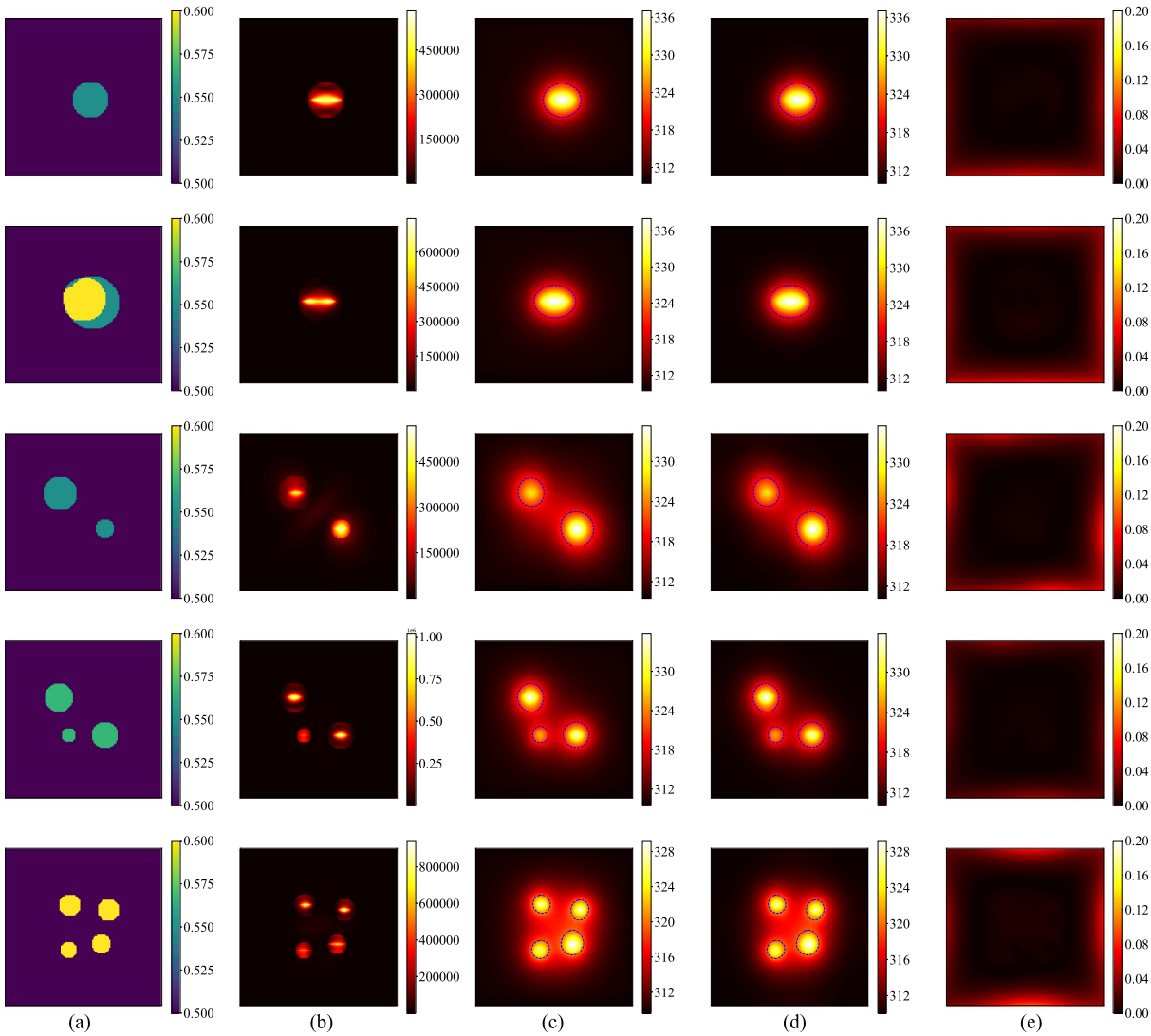


Fig. 9. Testing cases for SSUN, (a) thermal conductivity ($\mathrm{W/m/K}$), (b) dissipated power density ($\mathrm{W/m^3}$), (c) predicted temperature distribution (K), (d) accurate temperature distribution from FTCS (K) and (e) absolute error compared to the FTCS simulation (K). The dashed line in (c) and (d) is the 323.15 K isotherm.

choice of N and Δt_{div} for the TDUN is based on accuracy and efficiency considerations rather than stability, since the TDUN is not subject to the stability limit of the finite-difference based thermal solver. This is an appealing feature of this approach that adds to its computational efficiency. In the following, we show the results of the TDUN with $N = 10$ and $\Delta t_{div} = 1.5\Delta t_{FTCS}$.

During the training stage, 500 sets of training data are used to train the SSUN first. There is no overfitting based on the convergence of the training loss and testing loss. We double the number of training data to explore if the performance of the SSUN changes. It is observed that the training loss and testing loss have no difference compared to 500 training datasets. In the following, we report the results of the SSUN trained with 1000 sets of data and tested with 200 sets of data. The mean relative error of the whole testing set is 3.68×10^{-4} for the trained SSUN. Fig. 10 shows several testing cases of the SSUN. The per-case training time of the TDUN is much longer than the SSUN because each time step is trained during the training stage. So, we begin to train the TDUN with 60 sets of data and add another 60 sets to avoid overfitting. At last, the training loss and testing loss show that 240 sets of data are enough to train the TDUN. We test the performance of the trained TDUN with 200 sets of data. For the simulation of the heat diffusion by the TDUN, the mean relative error of each step is 5.33×10^{-4} on the testing set. The dynamic temperature evolution of the second, third and fourth cases in Fig. 9 is reported in Fig. 10.

V. Discussion

In this work, we proposed the TDUN and SSUN, physics-informed U-Nets that can predict the temperature distribution from the electromagnetic power density within inhomogeneous media and the material properties of these media. The accuracy of these two U-Nets has been demonstrated in Section IV. In this section, we will show the efficiency and generalization ability of this approach.

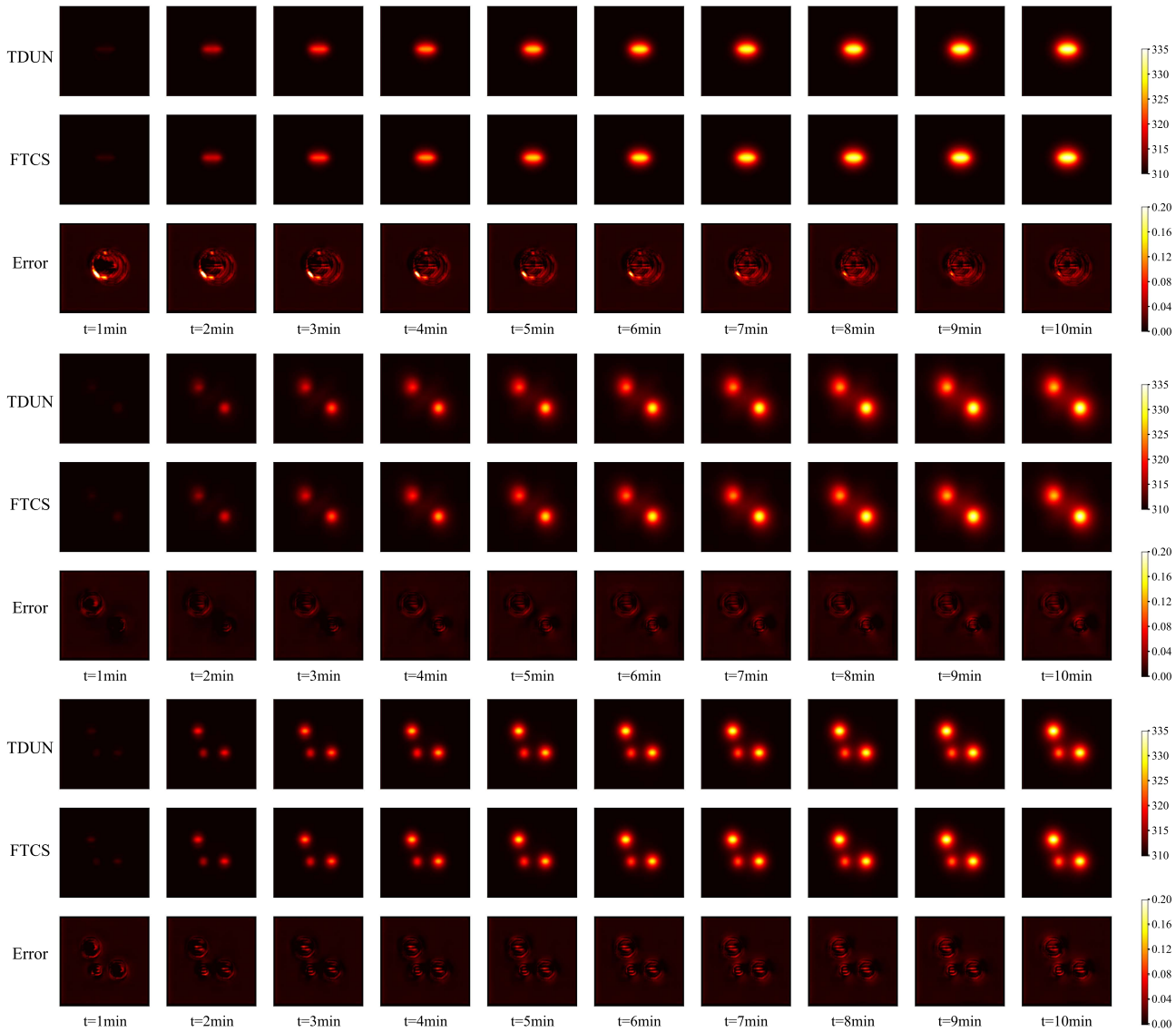


Fig. 10. Numerical solution of the heat equation from the TDUN; temperature (K) over time for the second, third and fourth case in Fig. 9, respectively, are shown. We also show the absolute error of the TDUN compared to the standard FTCS.

A. *Efficiency*

Scientific machine learning includes offline training and online prediction [18], [19], [20]. In the offline training stage, it took 1.01 hours to train the SSUN, and 3.08 hours to train the TDUN with $N = 10$ and $\Delta t_{div} = 1.5\Delta t_{FTCS}$ on one NVidia A100 GPU. However, the main advantage of scientific machine learning is that a trained network is faster than the numerical solver in the online prediction stage. Moreover, the proposed physics-informed approach is also superior to the supervised approach in the offline training stage because there is no need to spend time generating ground-truth data. In the following, we report the runtime for the TDUN and the SSUN on an Intel-i5 CPU and an NVidia GTX 3090 GPU. The runtime of the proposed U-Nets is compared to the runtime of the standard FTCS simulation. For FTCS, we only record the runtime of the algorithm, excluding the time used for initializing the simulation. For the U-Nets, the runtime accounts for the execution time of the SSUN or the TDUN.

It takes the trained SSUN 0.031 seconds on average to predict one steady-state temperature distribution on the CPU. However, FTCS takes 22.74 seconds to simulate one testing case on the same CPU. So, the speedup of the SSUN is about 538 times compared to the FTCS-based thermal solver. Besides, it takes 4.46 seconds on average to get the dissipated power density of the computational domain through FDTD. When the time used for the electromagnetic simulation is taken into account, the speedup is about 5 times compared to using FDTD coupled with FTCS for the whole electromagnetic-thermal simulation.

On an Intel-i5 CPU, a run of the TDUN takes 3.71×10^{-2} seconds with $N = 10$ and $\Delta t_{div} = 1.5\Delta t_{FTCS}$, which can replace 16.7 steps of FTCS. Each FTCS step takes 1.14×10^{-2} seconds, so the proposed TDUN is 5.24 times faster than the FTCS-based thermal solver. Besides, the runtime of the TDUN on a GPU is

TABLE III
COMPARISON OF EXECUTION TIME OF FTCS AND TDUN

	FTCS (s/case)	TDUN (s/case)
Intel-i5 CPU	22.74	4.34
NVidia GTX 3090 GPU	/	1.28

considered because convolutional neural networks take advantage of GPU computing. It takes the TDUN 1.1×10^{-3} seconds to predict one step of temperature distribution on an NVidia GTX 3090 GPU. The speedup is about 3.4 times compared to running the TDUN on the CPU. So, the proposed TDUN can be seen as an efficient GPU-based time-domain thermal solver to replace standard numerical solutions. We compare the runtime of applying the FTCS and the TDUN to simulate the evolution of the temperature to its steady state in Table III.

B. Generalization Ability

Unlike a supervised strategy [18], [19], [20], the scope of the TDUN and SSUN is not limited by the training data, but by the embedded heat equation and boundary conditions. The trained U-Nets can deal with any problem based on (20). In section IV, we use two-dimensional cases to show the feasibility of the proposed method for multiple, inhomogeneous objects within the computational domain. It is important to note that the electromagnetic source amplitude and the electromagnetic properties of the selected materials only affect the dissipated power density, which is normalized before entering the U-Nets. As a result, the TDUN and SSUN are, by construction, generalizable with respect to the electromagnetic sources involved. We can get a desired temperature profile by tuning the source. Then, the TDUN and SSUN are efficient tools for [2]. Moreover, the proposed method can be used to simulate other materials by changing the input material properties. Hence, it is generalizable to different materials.

VI. CONCLUSION

We showed the feasibility of applying physics-informed neural networks along with FDTD to simulate coupled electromagnetic-thermal problems. We proposed an unsupervised approach, which required no ground-truth data in the training stage. Two U-Nets, the TDUN and SSUN, were combined with the heat equation and boundary conditions to replace the thermal solver. They produced the dynamic evolution of temperature and the steady-state temperature distribution in the computational domain. Thus, electromagnetic-thermal problems were efficiently solved by FDTD (for the electromagnetic part of the simulation) combined with U-Nets (for the thermal part). Moreover, since U-Nets were trained in an unsupervised way, the time for generating ground-truth data was also reduced compared to supervised training. Numerical results have shown the accuracy, efficiency and generalization ability of the proposed method. This work is a first step towards three-dimensional PINN-based electromagnetic-thermal simulations. It also points to a new paradigm for solving multiphysics problems utilizing robust, single-physics techniques (such as FDTD or finite elements) combined with neural networks to model associated physics (such as mechanical, thermal and quantum effects).

REFERENCES

[1] F. Apollonio et al., "Mixed quantum-classical methods for molecular simulations of biochemical reactions with microwave fields: The case study of myoglobin," *IEEE Trans. Microw. Theory Tech.*, vol. 56, no. 11, pp. 2511–2519, Nov. 2008.

[2] S. Sharma and C. D. Sarris, "A novel multiphysics optimization-driven methodology for the design of microwave ablation antennas," *IEEE J. Multiscale Multiphys. Comput. Tech.*, vol. 1, pp. 151–160, 2016.

[3] F. Torres and B. Jecko, "Complete FDTD analysis of microwave heating processes in frequency-dependent and temperature-dependent media," *IEEE Trans. Microw. Theory Tech.*, vol. 45, no. 1, pp. 108–117, Jun. 1997.

[4] L. Ma et al., "Experimental validation of a combined electromagnetic and thermal FDTD model of a microwave heating process," *IEEE Trans. Microw. Theory Tech.*, vol. 43, no. 11, pp. 2565–2572, Nov. 1995.

[5] H. Huh and W. Kang, "Electrothermal analysis of electric resistance spot welding processes by a 3-D finite element method," *J. Mater. Process. Technol.*, vol. 63, no. 1/3, pp. 672–677, 1997.

[6] K. Wilczynski, M. Olszewska-Placha, and M. Celuch, "Application of conformal mapping to rigorous validation of 2D coupled EM-CFD modelling," in *Proc. IEEE/MTT-S Int. Microw. Symp.*, 2020, pp. 154–157.

[7] Z. Wei and X. Chen, "Deep-learning schemes for full-wave nonlinear inverse scattering problems," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 1849–1860, Apr. 2019.

[8] J. Xiao, J. Li, Y. Chen, F. Han, and Q. H. Liu, "Fast electromagnetic inversion of inhomogeneous scatterers embedded in layered media by born approximation and 3-D U-Net," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 10, pp. 1677–1681, Oct. 2020.

[9] W. Tang et al., "Study on a Poisson's equation solver based on deep learning technique," in *Proc. IEEE Elect. Des. Adv. Packag. Syst. Symp.*, 2017, pp. 1–3.

[10] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Rev. Phys.*, vol. 3, no. 6, pp. 422–440, 2021.

[11] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A deep learning library for solving differential equations," *SIAM Rev.*, vol. 63, no. 1, pp. 208–228, 2021.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2015, pp. 234–241.

[13] Y. Ge, L. Guo, and M. Li, "Physics-informed deep learning for time-domain electromagnetic radiation problem," in *Proc. IEEE MTT-S Int. Microw. Biomed. Conf.*, 2022, pp. 114–116.

[14] T. Shan, Z. Lin, X. Song, M. Li, F. Yang, and S. Xu, "A new approach for solving inverse scattering problems based on physics-informed supervised residual learning," in *Proc. 16th Eur. Conf. Antennas Propag.*, 2022, pp. 1–4.

[15] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[16] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.

[17] P. Karnakov, S. Litvinov, and P. Koumoutsakos, "Optimizing a DIscrete loss (ODIL) to solve forward and inverse problems for partial differential equations using machine learning tools," 2022, *arXiv:2205.04611*.

[18] S. Qi, Y. Wang, Y. Li, X. Wu, Q. Ren, and Y. Ren, "Two-dimensional electromagnetic solver based on deep learning technique," *IEEE J. Multiscale Multiphys. Comput. Tech.*, vol. 5, pp. 83–88, 2020.

[19] Y. Li et al., "Predicting scattering from complex nano-structures via deep learning," *IEEE Access*, vol. 8, pp. 139983–139993, 2020.

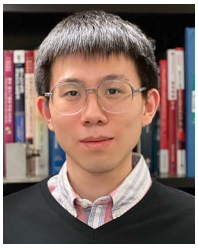
[20] Y. Wang, J. Zhou, Q. Ren, Y. Li, and D. Su, "3-D steady heat conduction solver via deep learning," *IEEE J. Multiscale Multiphys. Comput. Tech.*, vol. 6, pp. 100–108, 2021.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[22] H. H. Pennes, "Analysis of tissue and arterial blood temperatures in the resting human forearm," *J. Appl. Physiol.*, vol. 1, no. 2, pp. 93–122, 1948.

[23] Y. Cheng and M. Fu, "Dielectric properties for non-invasive detection of normal, benign, and malignant breast tissues using microwave theories," *Thoracic Cancer*, vol. 9, no. 4, pp. 459–465, 2018.

[24] J. Said Camilleri et al., "Review of thermal and physiological properties of human breast tissue," *Sensors*, vol. 22, no. 10, 2022, Art. no. 3894.
[25] A. Chanmugam, R. Hatwar, and C. Herman, "Thermal analysis of cancerous breast model," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, 2012, pp. 135–143.
[26] C. L. Brace, "Radiofrequency and microwave ablation of the liver, lung, kidney, and bone: What are the differences?," *Curr. Problems Diagn. Radiol.*, vol. 38, no. 3, pp. 135–143, 2009.
[27] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, vol. 114, no. 2, pp. 185–200, 1994.
[28] A. Karampatzakis, S. Kühn, G. Tsanidis, E. Neufeld, T. Samaras, and N. Kuster, "Antenna design and tissue parameters considerations for an improved modelling of microwave ablation in the liver," *Phys. Med. Biol.*, vol. 58, no. 10, 2013, Art. no. 3191.
[29] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, 2020, Art. no. 105524.



Shutong Qi (Graduate Student Member, IEEE) received the B.S. degree (with Hons.) in electronic and information engineering from Beihang University, Beijing, China, in 2020. He is currently working toward the Ph.D. degree in electrical engineering with the University of Toronto, Toronto, ON, Canada. From 2018 to 2020, he was a Research Assistant with the Institute of EMC Technology, Beijing, China. His research interests include scientific machine learning, computational electromagnetics, multiphysics simulation, numerical methods, and microwave circuit modeling.

Costas D. Sarris (Senior Member, IEEE) received the Diploma degree (with Hons.) in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 1997, the M.Sc. degree in applied mathematics from the University of Michigan, Ann Arbor, MI, USA, in 2002, and the Ph.D. degree in electrical engineering. He is currently a Full Professor with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. His research interests include computational electromagnetics, with an emphasis on time-domain modeling, adaptive mesh refinement, enhanced stability, and higher order methods. He also works on physics-based wireless propagation models (with full-wave, asymptotic, and hybrid techniques), uncertainty quantification, and scientific machine learning. He was the recipient of the IEEE Microwave Theory and Technology Society (MTT-S) Outstanding Young Engineer Award in 2013, Early Researcher Award from the Ontario Government in 2007, University of Toronto Faculty of Applied Science and Engineering Teaching Award in 2021, and four teaching awards for his work on undergraduate electromagnetics courses. He is the Co-Chair of the MTT-S Working Group on artificial intelligence (AI)/machine learning Based Technologies for Microwaves. He was the Technical Program Committee Chair of the 2015 IEEE Antennas and Propagation Society International Symposium on Antennas and Propagation and the Canadian National CommitteeUnited States National Committee Joint Meeting and the 2019 Numerical Electromagnetic and Multiphysics Modeling and Optimization Conference, the TPC Vice-Chair of the 2012 IEEE MTT-S International Microwave Symposium, and the Chair of the MTT-S Technical Committee on Field Theory and Numerical Electromagnetics from 2018 to 2020. He is the Editor-in-Chief of the IEEE JOURNAL ON MULTISCALE AND MULTIPHYSICS COMPUTATIONAL TECHNIQUES. He was a Guest Editor of the Special Issue on Machine Learning for Microwave Engineering of the *IEEE Microwave Magazine* in October 2021, and an Associate Editor for the IEEE MICROWAVE AND WIRELESS COMPONENTS LETTERS from 2007 to 2009 and the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES from 2009 to 2013.