

A Physics-Informed Machine Learning Approach for Heat Conduction Equation

Antons Cvečkovskis

(Dated: June 6, 2025)

This study presents a systematic investigation of Physics-Informed Neural Networks (PINNs) for solving steady-state and stationary heat conduction problems relevant to industrial applications. By embedding the governing heat equation and associated boundary conditions directly into the neural network loss function, PINNs provide a mesh-free and flexible alternative to conventional numerical methods such as the Finite Element Method (FEM). The research reviews recent advancements in PINN-based heat transfer modelling and implements both standard multilayer perceptron and Fourier feature-enhanced architectures. A range of benchmark problems—including 2D and 3D domains with Dirichlet, Neumann, and Robin boundary conditions—are solved and compared to FEM reference solutions. The results demonstrate that PINNs can accurately predict temperature distributions, achieving relative errors below 5% in most cases, with the choice of activation function and network architecture significantly impacting convergence and accuracy. Limitations in resolving sharp gradients near mixed boundary regions are analysed, and strategies for improvement are discussed. The findings highlight the potential of PINNs for rapid thermal analysis, digital twin creation, and integration into industrial design workflows.

I. INTRODUCTION

Over the last five years, Physics-Informed Machine Learning (PIML) has made noteworthy advancements as a powerful method for addressing issues governed by partial differential equations. PIML seeks to effectively forecast temperature distributions and thermal responses in intricate three-dimensional shapes by incorporating physical principles, like the heat equation, into machine learning frameworks. This approach is useful in industrial and academic contexts. For example, a CAD design of a heat sink part can be imported, boundary conditions can be specified, and a heat distribution can be predicted. Compared to conventional numerical solvers, PIML provides quicker evaluation for design iterations while still conforming to physical principles [1].

This paper uses a Physics-Informed Neural Network (PINN) approach to solve a steady-state heat transfer equation in 2 and 3 dimensions with Dirichlet, Neumann and Robin boundary conditions. The results are verified against the Finite Element Method (FEM).

Section II provides an overview of the recent developments in the field of PINNs in the context of heat transfer. Section III describes the mathematical background and numerical methods. In Section IV, we present and analyse the simulation results. Finally, Section V summarises the findings and provides outlooks for future work. The code is available on [GitHub](#).

II. RECENT APPLICATIONS IN HEAT TRANSFER

Physics-informed machine learning techniques are making significant strides in addressing a wide range of heat transfer challenges in both academia and industry. Here, we outline several examples that are directly applicable to manufacturing, materials processing, and thermal system design.

Authors of a 2021 paper, Zobeiry and Humfeld, [2] developed a PINN to solve the transient heat conduction in parts being heated inside industrial ovens. Their model included convective boundary conditions in the loss function. An adaptive normalising scheme was used to balance the partial differential equation (PDE) vs boundary condition (BC) losses, improving convergence. They also engineered input features based on heat transfer theory (e.g. non-dimensional temperature) to aid learning. The PINN was validated against finite element simulations for 1D and 2D cases, showing good accuracy. Notably, the trained model could predict thermal behaviour beyond the training zone and allowed fast what-if evaluations of different BCs, enabling feedback control loops for “Industry 4.0” smart manufacturing. This demonstrated the potential for near-real-time simulation of oven heating processes using PINNs.

In another interesting study [3], the authors addressed a large-scale 3D transient heat transfer problem related to the quenching process of aluminium alloy, which is a type of industrial heat treatment. They modelled a massive cylindrical workpiece (31 meters tall) placed in a furnace, accounting for varying convective and radiative heat fluxes on its surface. To solve this complex problem, they developed a PINN surrogate, referred to as MCO-PINN, which functions as a soft sensor to estimate the temperature field of the workpiece. This model was trained using the heat conduction PDE alongside a few thermocouple measurements. The authors encoded the varying convective and radiative boundary fluxes from different furnace zones into the loss function to accurately represent the physics involved. Additionally, they introduced a multi-loss consistency optimisation method in which each loss term—PDE, initial conditions, boundary conditions, and data—was treated as a probabilistic model. These were balanced through maximum likelihood estimation. To improve the model’s ability to capture localised thermal gradients, they employed Gaussian basis functions as activation functions in the net-

work. The resulting PINN demonstrated high accuracy in predicting the 3D temperature distribution within the alloy during the quenching process, achieving near real-time inference speeds. Industrial trials revealed that this PINN could track the workpiece temperature within a $\pm 3\text{ C}^\circ$ tolerance, which met the strict uniformity requirements while relying on only sparse sensor data. This case exemplifies a successful application of Physics-Informed Machine Learning in a production environment, providing a real-time thermal "digital twin" of the furnace process.

III. METHODS

A. Heat transfer equation

Any transient or steady heat-transfer problem must satisfy the law of conservation of energy. While thermodynamics tells us how much energy must be transferred to reach a new equilibrium, in engineering practice, the rate of heat transmission is often more critical than the total heat exchanged. Heat transfer is a key factor in the design of electronic devices, HVAC systems, power plants, and geothermal installations [4].

Heat conduction is the process by which thermal energy is transferred via microscopic interactions from regions of high temperature to regions of low temperature. In solids (and in fluids at negligible convective motion) this process is governed by Fourier's law. An energy balance on a differential control volume then yields the generic heat-conduction equation [4]:

$$\frac{\partial}{\partial x}\left(\kappa \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(\kappa \frac{\partial T}{\partial z}\right) + \dot{e}_{\text{gen}} = \rho c \frac{\partial T}{\partial t}, \quad (1)$$

where

- κ is the thermal conductivity,
- ρ is the density,
- c is the specific heat capacity,
- \dot{e}_{gen} is the volumetric heat-generation rate.

Introducing the thermal diffusivity $\alpha = \kappa/(\rho c)$ and dividing through by κ gives the Fourier–Biot form:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{\dot{e}_{\text{gen}}}{\kappa} = \frac{1}{\alpha} \frac{\partial T}{\partial t}. \quad (2)$$

At steady state the time derivative vanishes and one recovers the Poisson equation in Ω :

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{\dot{e}_{\text{gen}}}{\kappa} = 0. \quad (3)$$

Equivalently, in vector form,

$$-\nabla \cdot (\kappa \nabla T) = f \quad \text{in } \Omega, \quad (4)$$

where $f = \dot{e}_{\text{gen}}$.

B. Exploring the Theory Behind PINNs

In a PINN, one approximates the solution $u(x, y, z)$ by a neural network $u_\theta(x, y, z)$ with parameters θ . The heat transfer PDE must be supplemented with boundary conditions on the spatial domain's edges. The key idea is to train u_θ such that it satisfies the PDE and conditions as closely as possible [1] [5].

How does PINN incorporate the PDE? One defines a physics-informed loss function that penalises the PDE's residual. Sample a set of collocation points (also called training points) in the domain where one evaluates the PDE. These can be randomly chosen or a grid of (x, y, z) values in the domain (excluding the boundary for the PDE residual term). At each collocation point (x_i, y_i, z_i) , one computes the network's prediction $u_\theta(x_i, y_i, z_i)$ and then use automatic differentiation to compute the needed derivatives u_{xx}, u_{yy}, u_{zz} . The PDE residual is defined as

$$R(x, y, z) = f(x, y, z) - \kappa[u_{xx}(x, y, z) + u_{yy}(x, y, z) + u_{zz}(x, y, z)] \quad (5)$$

which should equal zero for an exact solution of the heat equation. The PINN loss will include a term that is the mean squared residual over all collocation points. For example, if we have N_f interior collocation points, the PDE loss term can be written as:

$$L_{\text{PDE}} = \frac{1}{N_f} \sum_{i=1}^{N_f} [f(x, y, z) - \kappa(u_{xx}(x, y, z) + u_{yy}(x, y, z) + u_{zz}(x, y, z))]^2 \quad (6)$$

which drives the neural network to satisfy $u_t \approx \kappa(u_{xx} + u_{yy} + u_{zz})$ at those points. All derivatives here are obtained via automatic differentiation through the neural network's computation graph, which is a built-in capability of modern deep learning frameworks. This means we do not need to derive finite difference formulas. The network can be differentiated analytically, providing a mesh-free way to evaluate PDE terms at arbitrary points.

How are boundary and initial conditions enforced? PINNs incorporate these by adding additional loss terms. We sample points on the boundaries of the domain and penalise the deviation from the known conditions. For instance, suppose we have Dirichlet boundary conditions specifying $u(x, y, z)$ on the domain boundaries. We can define boundary loss

$$L_{\text{BC}} = \frac{1}{N_b} \sum_{j=1}^{N_b} \left(u_\theta(x_j^b, y_j^b, z_j^b) - u_{\text{BC}}(x_j^b, y_j^b, z_j^b) \right)^2, \quad (7)$$

where (x_j^b, y_j^b, z_j^b) are points on the boundary and u_{BC} is the given boundary value. This term ensures the network outputs the correct temperature on the boundaries.

If the boundary conditions are Neumann or Robin, a similar approach is taken except we penalise the derivative (computed by auto-diff) minus the specified value. The loss terms are formulated accordingly. For example, for a Neumann boundary $u_n = g(x, y, z)$, one would include $(\partial u_\theta / \partial n - g)^2$ in L_{BC} for points on that boundary. Finally, we combine all these contributions into a single total loss that the network will minimise:

$$L_{\text{Total}} = L_{\text{PDE}} + L_{\text{BC}} + L_{\text{Data}} \quad (8)$$

where we have also included L_{data} to represent any reference solution data or observational data from the domain. The L_{data} term would be a mean squared error between the network prediction and known solution values at a set of data points in the domain. In many cases where no additional interior data is available, L_{data} is omitted and the PINN is entirely self-supervised by $L_{\text{PDE}} + L_{\text{BC}}$ – making it an unsupervised learning approach guided by physics.

During training, the PINN adjusts its weights θ to minimise L_{total} . This is done with standard optimisation algorithms like gradient descent or Adam, and the gradients of the loss with respect to θ are computed by backpropagation. Importantly, the backpropagation naturally integrates the PDE constraint because the PDE residual term itself uses gradients of the network, thanks to auto-differentiation.

C. Fourier Network

In the Fourier Network (F-Net) architecture, the standard feed-forward network $u_\theta(\mathbf{x})$ is replaced by a composition of a fixed Fourier feature embedding $\Phi(\mathbf{x})$ and a base multilayer perceptron [6]. This enables efficient representation of high-frequency modes in the solution, which is particularly useful for capturing steep gradients in heat conduction problems such as those encountered in heat sinks.

Let $\mathbf{x} = (x, y, z) \in \Omega \subset \mathbb{R}^3$ denote a spatial point. We define the Fourier feature map

$$\Phi(\mathbf{x}) = [\sin(2\pi \mathbf{B} \mathbf{x}), \cos(2\pi \mathbf{B} \mathbf{x})] \in \mathbb{R}^{2m}, \quad (9)$$

where $\mathbf{B} \in \mathbb{R}^{m \times 3}$ is a fixed matrix of frequencies, typically sampled once from a Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$. The hyperparameter σ controls the bandwidth of the embedding and thus the highest spatial frequencies the network can represent.

The Fourier Network predicts the temperature field via

$$u_\theta(\mathbf{x}) = f_\theta(\Phi(\mathbf{x})), \quad (10)$$

where $f_\theta : \mathbb{R}^{2m} \rightarrow \mathbb{R}$ is an MLP with parameters θ . A typical choice is

$$f_\theta(\mathbf{z}) = W^{(L)} \sigma(W^{(L-1)} \sigma(\cdots \sigma(W^{(1)} \mathbf{z} + b^{(1)}) \cdots) + b^{(L-1)}) + b^{(L)}, \quad (11)$$

with $\sigma(\cdot)$ denoting an element-wise activation tanh or ReLU¹, for example. Depth L and width of each hidden layer are selected based on the complexity of the thermal gradients in the heat sink geometry.

Training proceeds similarly to standard PINNs: we minimise a composite loss

$$L_{\text{Total}} = L_{\text{PDE}} + L_{\text{BC}} + L_{\text{Data}}, \quad (12)$$

but with $u_\theta(\mathbf{x})$ now defined by the Fourier Network. In particular, the PDE residual at an interior collocation point \mathbf{x}_i is

$$R(\mathbf{x}_i) = f(\mathbf{x}_i) - \kappa(u_{xx}(\mathbf{x}_i) + u_{yy}(\mathbf{x}_i) + u_{zz}(\mathbf{x}_i)),$$

$$u_{pq} = \frac{\partial^2}{\partial p \partial q} u_\theta(\mathbf{x}), \quad p, q \in \{x, y, z\}, \quad (13)$$

and

$$L_{\text{PDE}} = \frac{1}{N_f} \sum_{i=1}^{N_f} [R(\mathbf{x}_i)]^2. \quad (14)$$

All derivatives u_{pq} are computed by automatic differentiation of $f_\theta \circ \Phi$.

Boundary conditions are enforced as before. For Dirichlet boundaries at points $\{\mathbf{x}_j^b\}$,

$$L_{\text{BC}} = \frac{1}{N_b} \sum_{j=1}^{N_b} [u_\theta(\mathbf{x}_j^b) - u_{\text{BC}}(\mathbf{x}_j^b)]^2, \quad (15)$$

and Neumann or Robin conditions are handled by penalising normal derivatives $\partial u_\theta / \partial n$ accordingly.

D. Variational formulation

The variational formulation of the heat conduction can be found in Appendix V.

Tools

We utilised the Python library `matplotlib` [7] to create all illustrations in this report. All PINN calculations were performed using the NVIDIA `PhysicsNeMo` Python library for deep learning [8]. All FEM results were calculated using the `FenicsX` Python library [9].

IV. RESULTS AND DISCUSSION

The exponential learning rate loss was used in all numerical experiments due to the best convergence speed and accuracy.

¹ Rectified Linear Unit, defined as $f(x) = \max(0, x)$

A. First Steps - Heat Conduction with Dirichlet BC

Let's start with a unit square domain Ω with Dirichlet boundary conditions. The left boundary is kept at the $T_{\text{left}} = 25^\circ\text{C}$, while the rest of the boundaries are kept at the $T = 5^\circ\text{C}$.

The finite-element solution on a 201×201 structured mesh is used as reference throughout this section. For every experiment, we monitor the aggregated training loss² and the relative \mathcal{L}_2 -error

$$\mathcal{L}_{2,\text{rel}} = \frac{\|T_{\text{PINN}} - T_{\text{FEM}}\|_2}{\|T_{\text{FEM}}\|_2},$$

computed on 10^4 uniformly spaced evaluation points inside Ω .

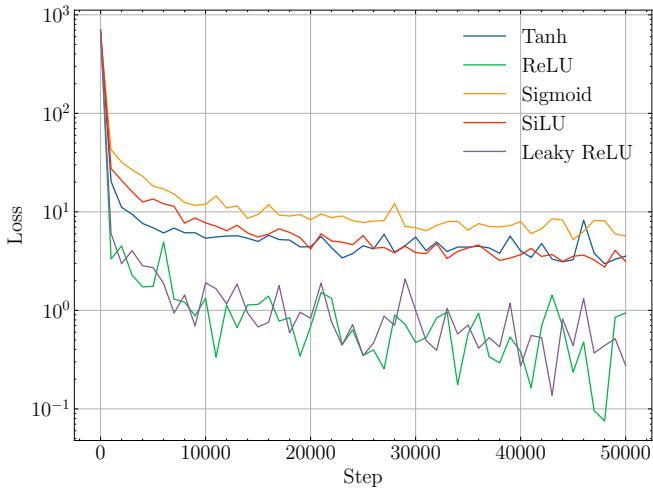


FIG. 1. Aggregated loss for the study of the effect of different activation functions for 3 hidden layers of width 128. ReLU shows the best result, however, it also produces the largest $\mathcal{L}_{2,\text{rel}}$ across all activation functions. The compromise is tanh or ReLU activation function.

Figures 1 and 2 show that tanh and ReLU exhibit both the fastest initial loss decay and the lowest final $\mathcal{L}_{2,\text{rel}}$ ($\approx 3 \times 10^{-2}$ at 50k iterations). ReLU suffers from piecewise-linearity, producing a plateau at $\mathcal{L}_{2,\text{rel}} \approx 9 \times 10^{-1}$ despite a seemingly decreasing total loss. Sigmoid saturates early, while SiLU³ improves upon ReLU yet still underperforms tanh/ReLU by roughly a factor of 2 in error. The observation confirms that smooth, non-saturating activations with non-zero higher derivatives facilitate the automatic differentiation required for accurate PDE residuals.

Increasing expressive power (Figures 3 and 4) produces the expected monotonic trend: the widest and

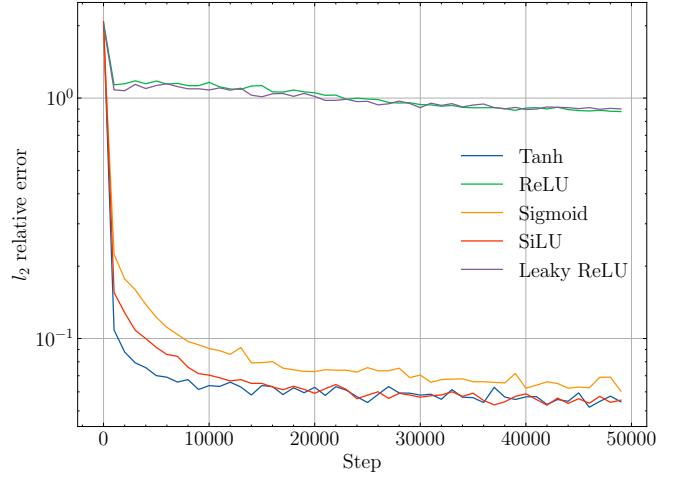


FIG. 2. $\mathcal{L}_{2,\text{rel}}$ for experiments with different activation functions for 3 hidden layers of width 128. tanh and ReLU produce the best results.

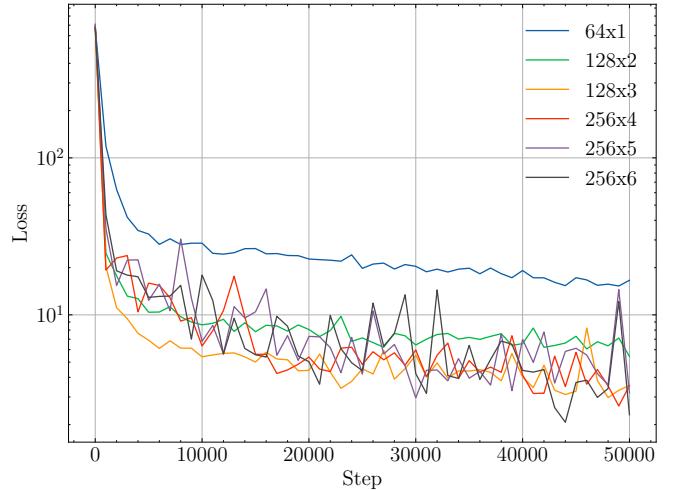


FIG. 3. Aggregated loss for six architectures. The lowest training error 5×10^{-2} is produced by the widest and deepest network with 256 neurons across 6 hidden layers.

deepest model (256 neurons, 6 hidden layers) attains the lowest error ($\mathcal{L}_{2,\text{rel}} \approx 5 \times 10^{-2}$), close to the mesh-converged FEM benchmark. However, diminishing returns are already visible: the jump from 256×4 to 256×6 reduces the error by only $\sim 5\%$ but increases the wall time by $\sim 65\%$. A moderate configuration (128×3) therefore offers a good compromise between accuracy and cost for subsequent physics experiments.

a. *Interior collocation point amount.* Figures 5 and 6 now compare five interior batches $N_f = 100, 500, 1000, 2000, 4000$ for the baseline 128×3 tanh network. A few clear trends emerge:

- With only 100 collocation points, the optimiser receives an impoverished picture of the PDE residual, leading to an initial loss plateau two orders of

² Sum of PDE and boundary condition data terms.

³ The smooth-ReLU or *swish* activation.

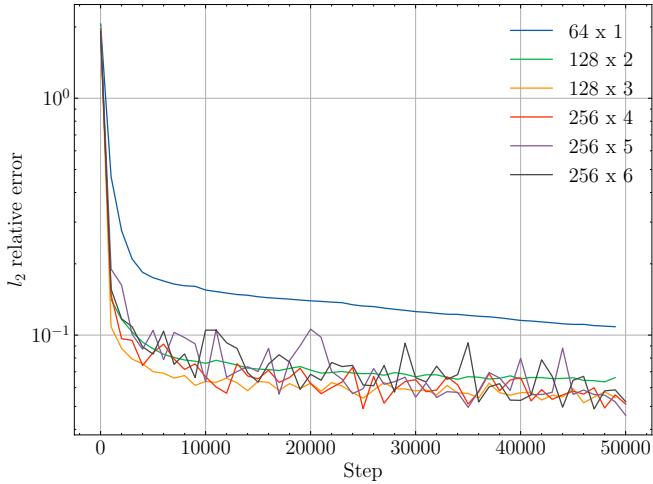


FIG. 4. $\mathcal{L}_{2,\text{rel}}$ for six architectures. The best result is produced by the deepest network - (256 neurons, 6 hidden layers), it attains the lowest error ($\mathcal{L}_{2,\text{rel}} \approx 5 \times 10^{-2}$).

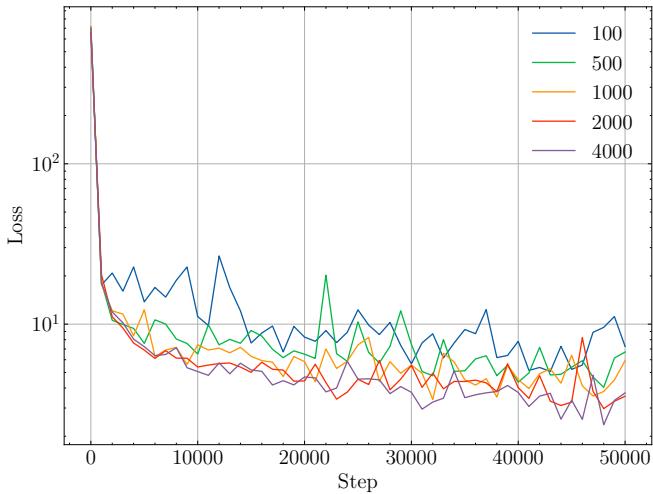


FIG. 5. Aggregated loss for 100, 500, 1000, 2000 and 4000 internal collocation points, respectively. Smaller batches arrive at a larger aggregated loss at the end of the training, while larger batches ($N \geq 1000$) converge faster and get a smaller error of the loss function.

magnitude higher than the richer samplings and a visibly noisier gradient signal. The relative error drops below 10^{-1} around $\sim 4 \cdot 10^4$ iterations.

- Increasing N_f from 500 to 1000 sharply improves stability. All three curves ($N_f \geq 1000$) collapse onto the same trajectory after the first 10^4 iterations, indicating that the network has already resolved the dominant diffusion modes.
- Beyond 1000 points doubling to 2000 and again to 4000 yields *no measurable gain* in either aggregated loss or the final $\mathcal{L}_{2,\text{rel}}$.

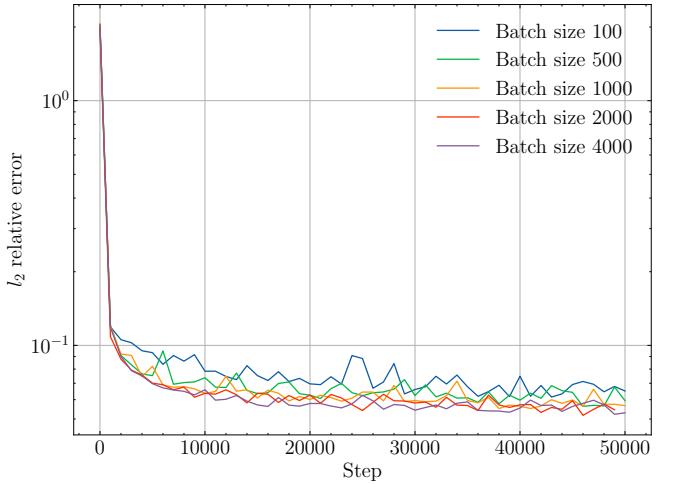


FIG. 6. Effect of the interior collocation point amount (\tanh , 128×3 network) on $\mathcal{L}_{2,\text{rel}}$. By increasing the collocation point size, one gets a smaller $\mathcal{L}_{2,\text{rel}}$ relative error, as expected. However, doubling the batch size after 1000 points doesn't increase the convergence significantly, meaning that there exists a balance between accuracy and computation time. This compromise is 1000 or 2000 points for this particular problem.

b. Correlation between loss and error. Across all experiments, the evolution of the aggregated loss mirrors that of $\mathcal{L}_{2,\text{rel}}$ after an initial ‘burn-in’ of roughly 5×10^3 iterations. Nonetheless, activation functions with poor conditioning (e.g. sigmoid) occasionally achieves a low total loss while retaining a high relative error, underlining that *balanced* loss terms and proper activation choice is crucial.

B. Heat Conduction with both Radiative and Convective BC

Next, let's look at a more complicated experiment. The heat conduction test from [10] inspired this problem. The test case illustrates a stationary thermal analysis in two dimensions. The model consists of a ceramic strip within a material with thermal conductivity $\kappa = 3 \text{ W/m} \cdot \text{K}$.

The side edges of the strip are kept at a constant temperature $T_{\text{hot}} = 1173 \text{ K}$. The upper surface of the strip emits heat through both convection and radiation to the surrounding environment at $T_{\text{amb}} = 323 \text{ K}$. The lower boundary is considered to be thermally insulated.

The problem was non-dimensionalised using the dimensionless temperature $\theta = \frac{T - T_{\text{amb}}}{\Delta T}$ and the effective Biot number, so the strong formulation of the problem becomes

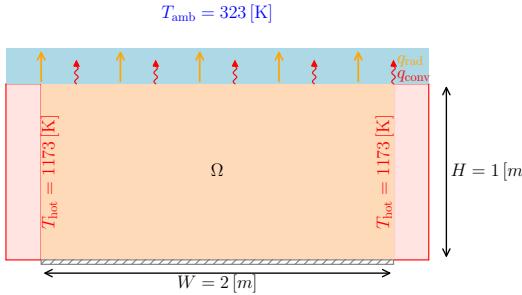


FIG. 7. Geometry for heat transfer in a domain Ω with $\kappa = 3 \text{ W/m} \cdot \text{K}$. The top boundary is both convective and radiative with ambient temperature $T_{\text{amb}} = 323 \text{ K}$. The right and left walls are maintained at a constant temperature $T_{\text{hot}} = 1173 \text{ K}$, and the bottom wall is thermally insulated.

$$\frac{\partial^2 \theta}{\partial \hat{x}^2} + \left(\frac{L}{H}\right)^2 \frac{\partial^2 \theta}{\partial \hat{y}^2} = 0 \text{ in } \Omega \quad (16)$$

$$\theta = 1 \text{ on } \Gamma_{\text{left/right}} \quad (17)$$

$$\frac{\partial \theta}{\partial \hat{y}} = 0 \text{ on } \Gamma_{\text{bottom}} \quad (18)$$

$$\frac{\partial \theta}{\partial \hat{y}} = \text{Bi}_{\text{eff}} \text{ on } \Gamma_{\text{top}}. \quad (19)$$

To get back the physical temperature, the output node is scaled back by $T = \theta \cdot \Delta T + T_{\text{amb}}$.

The Neural Network solver was run for 100,000 iterations, with the best convergence speed and accuracy found with 5 hidden layers, 256 in size, with the SiLU activation function.

The left y axis in figure 8 illustrates the evolution of the individual loss components—namely, the diffusion residual, convective boundary, and insulated boundary losses—as well as their aggregated sum over the course of 100,000 training steps. We observe a rapid initial decrease in the diffusion-residual loss, which drops to approximately 10^{-4} within the first 20,000 steps, followed by a more gradual but consistent decline in the following iterations. The convective and insulated boundary losses show similar behaviour, indicating that the PINN effectively enforces both the internal PDE and the external boundary conditions. By around 80,000 steps, the aggregated loss plateaus at about 10^{-3} , suggesting that the solution representation converges.

We compared our result with the weak solution given by the Finite Element Method (see Figure 8 right axis). Around 15,000 steps, the l_2 relative error falls below 0.15 and stays approximately the same throughout the learning process.

Comparing the pointwise difference between the weak solution and our result (see Figure 15), the PINN accurately reproduces the uniform hot side-edge temperature and the smooth temperature gradient toward the convective-radiative top surface. The maximum absolute

error remains below 3 K in most of the interior but increases near the top-corner regions up to 200 K in some points, where the combined effect of steep gradients and mixed boundary conditions is most challenging, meaning that the solution is accurate inside the domain, but gets inaccurate near the top-corner edges. We tried to apply the Spatial Weighting of Losses given by $y = 1 - |x|$ weighing function near the top-corner edges to minimise the error and avoid sharp discontinuity, but the results were inconclusive, meaning that further research is necessary to obtain better and smoother results.

Potential improvements can be through localised collocation point enrichment and different adaptive weighting of boundary losses.

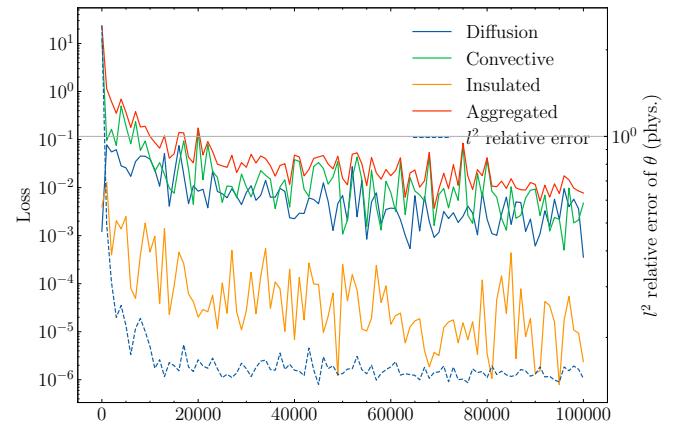


FIG. 8. On the left y axis - training convergence of the PINN for the 2D steady-state heat conduction problem. Shown are the individual loss for interior PDE, convective–radiative boundary residual (top surface), and Dirichlet boundary residual (hot side edges), Insulated boundary and their aggregated sum as functions of training iterations. Rapid initial decay followed by a gradual plateau indicates balanced enforcement of governing equations and boundary conditions. On the right y-axis - evolution of the relative \mathcal{L}_2 error over the course of training for the 2D steady-state heat conduction problem. The error falls below 0.15 around 15,000 iterations, demonstrating convergence against the FEM reference solution.

C. Heat Box

The geometry in this experiment is a $2 \times 2 \times 2$ box with a heat source on the bottom and convective walls. The 0.5×0.5 central region of the bottom plate, the heat source, is prescribed to a Neumann flux of 100 K/m , while the rest of the bottom boundary is insulated. The rest of the walls are convective with $h = 50 \text{ W}/(\text{m}^2 \cdot \text{K})$. The thermal conductivity is set to $\kappa = 3 \text{ W} \cdot \text{m}^{-1} \text{K}^{-1}$. For smoother learning, the heat source was smoothed by a tanh step function. This experiment mimics, for example, the centralised hot spot of a packaged power device mounted inside a cooled casing. This experiment can be

considered as a stepping stone for the next experiment, described in the next section, - Heat Sink.

Three models were tested:

1. a Fourier Network with 5 hidden layers, each 128 in width and 10 sinusoidal features using frequencies along each coordinate axis (**F-Net 128**),
2. a Fourier Network with 5 hidden layers, each 512 in width and 10 sinusoidal features using frequencies along each coordinate axis (**F-Net 512**),
3. a conventional fully-connected multilayer perceptron of 5 hidden layers with width 128 (**FC 128**).

All architectures employed five hidden layers, the SiLU activation ⁴, the Adam [11] optimiser, and the exponential learning-rate scheduler ($\eta_0 = 10^{-3}$, $\gamma = 0.95$, step = 8000). Interior, boundary, and source constraints had 5000 / 2000 / 2000 points per batch, respectively. Training was stopped after 3×10^5 iterations. Figure 9 compares the aggregated loss histories, while Fig. 10 shows the corresponding relative \mathcal{L}_2 errors with respect to a high-resolution FEM solution on 120^3 hexahedral elements.

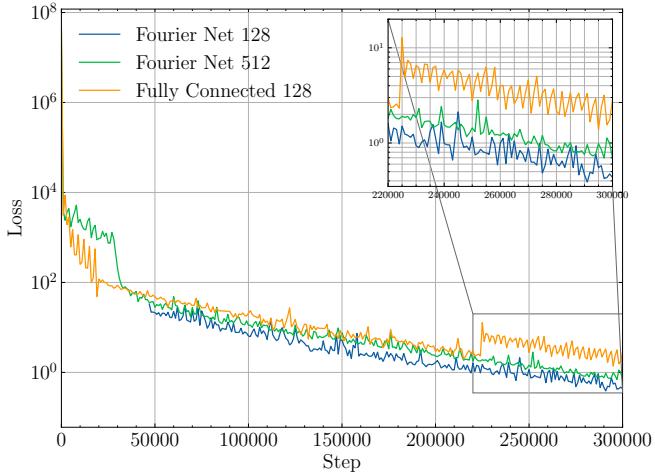


FIG. 9. Aggregated training loss for Fourier Networks with 128 and 512 hidden layer sizes and Fully Connected Neural Network with 128 hidden layer size, each with 5 hidden layers. The best performance is by the Fourier network 128. The oscillations in Fully Connected NN can be explained by sharp gradient changes as it tries to fit the sharp heat source flux.

To shed more light on the different convergence patterns, we monitored the maximum absolute component of the parameter gradient throughout training (Fig. 11). Both Fourier-Net variants exhibit a smooth, almost monotonic decay of $\max |\nabla_\theta L|$ from $\sim 10^{-1}$ to $\sim 10^{-2}$, indicating that their weight updates gradually enter the

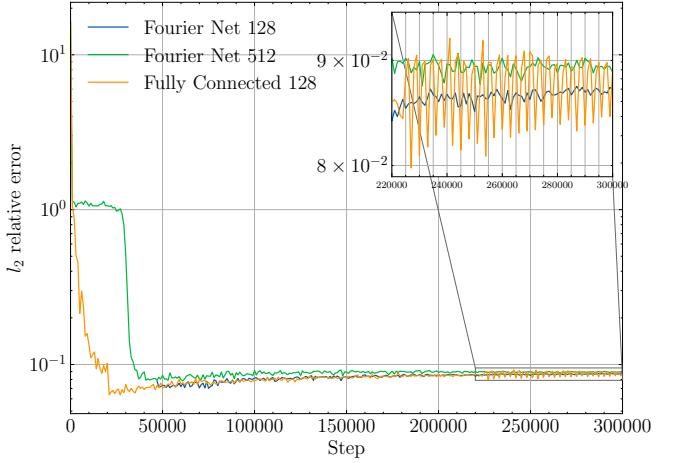


FIG. 10. Relative \mathcal{L}_2 error versus training step. Fourier and fully connected networks converge approximately at the same rate and converge to the same values; alas, the oscillations can be observed for the fully connected network.

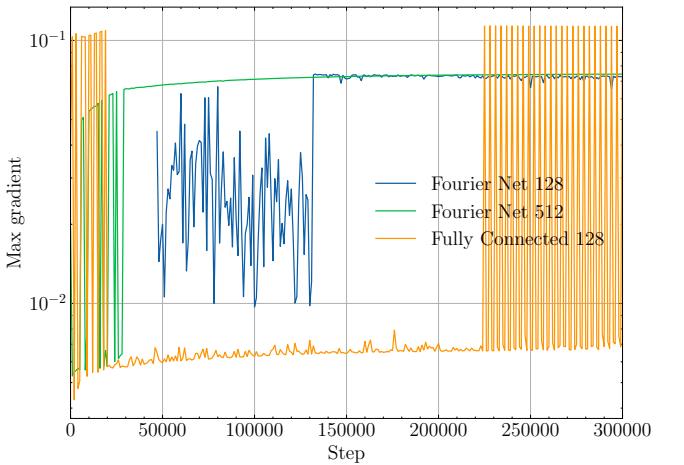


FIG. 11. Evolution of the maximum absolute parameter gradient for the three network variants. Large oscillations can be observed for the fully connected network, while Fourier networks exhibit smoother gradients.

small-gradient regime once the dominant diffusion modes are resolved. In contrast, the fully connected network shows pronounced high-frequency oscillations during the first 5×10^4 steps as the optimiser attempts to reconcile the sharp, localisation-induced gradients near the square heat source, with a much smoother field elsewhere. These gradient spikes correlate with the loss overshoots (orange curve in Fig. 9) and translate into the error ‘‘ringing’’ visible in Fig. 9.

The **F-Net 128** model consistently attains the lowest loss and converges to a final error of $\mathcal{L}_{2,\text{rel}} \approx 8.5 \times 10^{-2}$. The wider **F-Net 512** learns a bit more rapidly settling near 9.0×10^{-2} . The fully-connected baseline suffers from an initial error spike (overshoot caused by large gradients near the heat-source perimeter) yet eventually reaches a

⁴ Sigmoid function multiplied by its input, or $x\sigma(x)$.

comparable accuracy ($\sim 8.8 \times 10^{-2}$). These observations reinforce the benefit of Fourier feature embeddings for capturing the steep local gradients inside a diffusive field.

To visualise the learned temperature field, two slices are compared against the FEM benchmark in Figs. 16 and 17. Both slices confirm that the PINN reproduces the quasi-parabolic decay from the heat spot and the symmetry of the solution. The maximum absolute error stays below 0.8 C° throughout the domain, with the largest deviations occurring (i) along the sharp edges of the square flux patch and (ii) near the top corners where the convective boundary intersects the adiabatic bottom, a region characterised by secondary gradients that are undersampled by the default collocation strategy.

Although the present error levels are already below 1% of the peak temperature rise, further refinement is possible. Preliminary tests with (i) adaptive collocation enrichment inside the heat-source neighbourhood and (ii) importance weighting of the Robin residual reduced the edge artefacts but increased training time substantially.

D. Heat Sink

The final experiment is a heat sink with 5 fins (see Figure 12), the parameters of the geometry can be found in Table I. The heat source on the bottom face is simulated with a Neumann boundary condition with a value of 100 K/m . Here, we used the same smoothing function described in the previous section to smooth the flux of the heat source. The fins are convective with $h = 1\text{ W/(m}^2 \cdot \text{K)}$ and the conductivity is set again to $\kappa = 3\text{ W} \cdot \text{m}^{-1} \text{K}^{-1}$. The ambient temperature is 30 C° . A volumetric heat source $f = 1\text{ W/m}^3$ is also present in the domain.

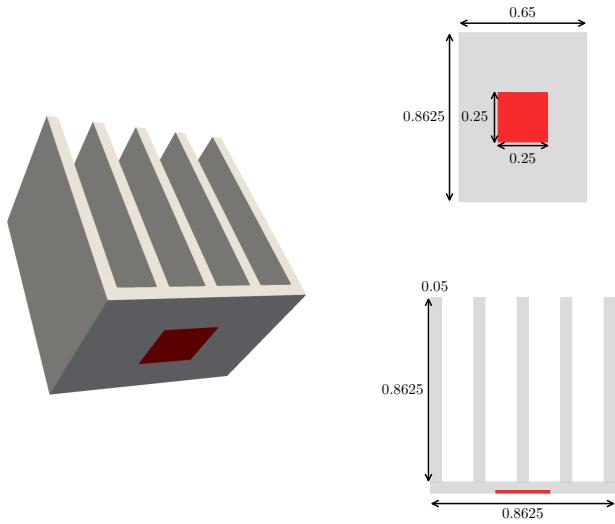


FIG. 12. Heat sink geometry

This geometry is difficult to simulate because of fins that create sharp gradients, making it challenging for a

regular fully connected neural network to learn the solution effectively because of slow convergence. This highlights the advantage of the Fourier feature embedding for representing high-frequency gradients. The optimal number of frequencies varies with each problem, requiring a balance between accuracy benefits and the computational costs incurred by utilising additional Fourier features. In this experiment, we compared the MLP with 5 hidden layers with size 128 and Fourier network with the same hidden layer configuration and 128 frequencies sampled from the Gaussian distribution with zero mean and standard deviation $\sigma = 2$.

Dimension	Value
Heat source ($1 \times w$)	0.25×0.25
Base plate ($1 \times w \times h$)	$(0.65 \times 0.8625 \times 0.05)$
Fin ($1 \times w \times h$)	$(0.65 \times 0.0075 \times 0.8625)$

TABLE I. Heat sink dimensions

Unfortunately, due to the limited nature of time, we were unable to fine-tune the models and allow for their complete convergence. A significant amount of time was spent identifying the optimal combination of parameters that we hope will yield the best results. Nevertheless, let's examine the intermediate results.

To assess the learning behaviour and convergence, we tracked the evolution of the individual loss components—namely, the diffusion, bottom plate, and convective boundary losses—for both the fully connected network and the Fourier network architectures (see Figures 18 and 19). We can see that the bottom plate is converging faster than the others because of the applied smoothing. The most difficult part is the diffusion itself, which can be seen as large spikes across the training period. Figures 13 and 14 present the aggregated training losses and $\mathcal{L}_{2,\text{rel}}$ error against a fine-meshed FEM solution for the fully connected and Fourier feature networks, respectively. Notably, both models show rapid initial decay of the loss terms, followed by a gradual plateau as training progresses. The Fourier network exhibits slightly smoother convergence behaviour, which is attributed to its improved capability to represent high-frequency solution components in regions with steep gradients. The predicted temperature fields for both networks in the xy and xz planes can be seen in Figures ??, 21 and 22, 23. We note that the temperature fields look physical, however, due to not full convergence, the difference between PINNs and FEM is above 6 C° for fully connected network and above 4 C° for the Fourier. Further training will produce more accurate temperature representations and the $\mathcal{L}_{2,\text{rel}}$ error will decrease and go to 0.

Overall, the results indicate that the Fourier network outperforms the standard multilayer perceptron in this challenging heat sink problem, both in terms of training stability and solution accuracy. Nonetheless, the observed errors near boundary intersections and in regions with mixed boundary conditions suggest that further

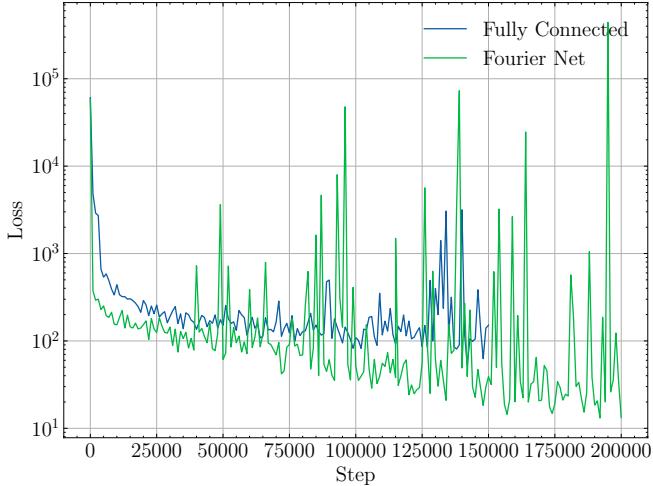


FIG. 13. Aggregated loss for fully connected and Fourier network architectures over the 150000 and 200000 training iterations, respectively.

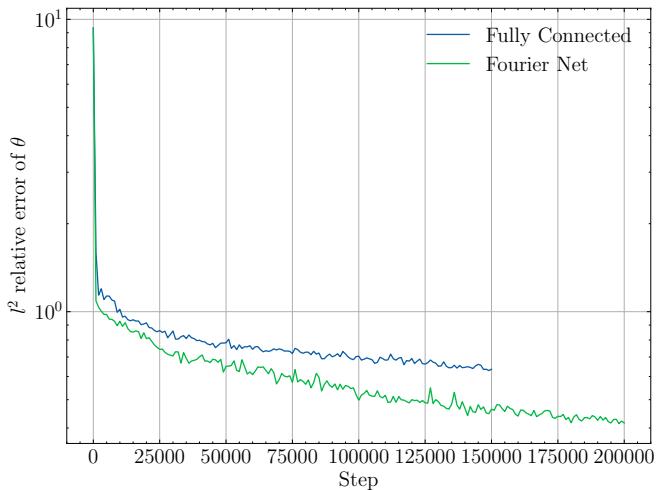


FIG. 14. Relative $\mathcal{L}_{2,\text{rel}}$ error for fully connected and Fourier network architectures over the 150000 and 200000 training iterations, respectively, evaluated against the FEM reference solution. The Fourier network converges more quickly.

improvements are possible. Potential future enhancements include adaptive collocation point refinement and localised loss weighting, particularly in problematic regions where the current PINN approach struggles to resolve steep temperature gradients.

V. CONCLUSION

In this study, we present a comprehensive investigation of Physics-Informed Neural Networks used to solve steady-state and stationary heat conduction problems relevant to industrial applications. By systematically embedding the governing heat equation and various bound-

ary conditions into the loss function of neural networks, we demonstrate that PINNs offer a mesh-free, flexible, and accurate alternative to classical numerical methods, such as the Finite Element Method.

Our results indicate that PINNs, particularly when enhanced with Fourier feature embeddings and modern activation functions, can achieve relative errors below 5% when compared to high-resolution FEM solutions. The study further reveals the influence of network architecture, activation function, and the selection of collocation points on both convergence rate and solution accuracy. While standard multilayer perceptrons perform well in smooth regions, Fourier feature networks show superior performance in capturing steep gradients and localised heat sources.

A detailed analysis highlights both the strengths and current limitations of the PINN approach. Notably, PINNs can effectively enforce complex boundary conditions, including Dirichlet, Neumann, and Robin types, and generalise well across various geometries. However, challenges persist in accurately resolving sharp gradients and mixed boundary regions, where solution errors tend to concentrate. Our preliminary attempts at spatial weighting and boundary enrichment suggest promising directions, but we recommend exploring more robust strategies, such as adaptive collocation and importance sampling, for future work.

From an industrial perspective, the potential of PINNs for rapid thermal analysis and digital twin creation is significant. Their ability to provide fast, mesh-free solutions paves the way for integration into real-time design and control workflows, particularly in advanced manufacturing and electronics cooling.

Future work should focus on extending PINNs to time-dependent heat conduction problems, exploring temperature-dependent material properties, and further developing adaptive sampling and hybrid data-driven approaches. Additionally, scaling PINNs to larger and more complex three-dimensional industrial geometries while maintaining computational efficiency remains an open challenge. Continued research in these areas will further strengthen the applicability of PINNs as a next-generation tool in computational heat transfer and other multiphysics domains.

APPENDIX A: THE WEAK FORMULATION

In this chapter, we derive the variational (weak) form of the heat conduction equation, incorporating Neumann and Robin boundary conditions, and identify the corresponding bilinear and linear forms.

Starting from the strong form

$$-\nabla \cdot (\kappa \nabla u) = f \quad \text{in } \Omega, \quad (20)$$

we multiply by a test function v and integrate over Ω , then integrate by parts:

$$-\int_{\Omega} \nabla \cdot (\kappa \nabla u) v \, dx = \int_{\Omega} \kappa \nabla u \cdot \nabla v \, dx - \int_{\partial\Omega} \kappa \frac{\partial u}{\partial n} v \, ds. \quad (21)$$

Split the boundary $\partial\Omega$ into Neumann parts Γ_N^i and Robin parts Γ_R^i . Then

$$\begin{aligned} -\int_{\partial\Omega} \kappa \frac{\partial u}{\partial n} v \, ds &= -\sum_i \int_{\Gamma_N^i} \kappa \frac{\partial u}{\partial n} v \, ds - \\ &\quad \sum_i \int_{\Gamma_R^i} \kappa \frac{\partial u}{\partial n} v \, ds. \end{aligned} \quad (22)$$

On Neumann boundaries, we prescribe $\kappa \partial_n u = g_i$ on Γ_N^i , and on Robin boundaries $\kappa \partial_n u + r_i u = r_i s_i$ on Γ_R^i . Hence

$$\begin{aligned} -\int_{\partial\Omega} \kappa \frac{\partial u}{\partial n} v \, ds &= \sum_i \int_{\Gamma_N^i} g_i v \, ds \\ &\quad + \sum_i \int_{\Gamma_R^i} r_i (u - s_i) v \, ds. \end{aligned} \quad (23)$$

Combining (21) and (23), the full variational form reads:

$$\begin{aligned} F(u, v) &= \int_{\Omega} \kappa \nabla u \cdot \nabla v \, dx + \sum_i \int_{\Gamma_R^i} r_i (u - s_i) v \, ds + \\ &\quad \sum_i \int_{\Gamma_N^i} g_i v \, ds - \int_{\Omega} f v \, dx = 0. \end{aligned} \quad (24)$$

To identify the bilinear and linear pieces, note on each Robin boundary

$$\int_{\Gamma_R^i} r_i (u - s_i) v \, ds = \int_{\Gamma_R^i} r_i u v \, ds - \int_{\Gamma_R^i} r_i s_i v \, ds. \quad (25)$$

Thus we set

$$a(u, v) = \int_{\Omega} \kappa \nabla u \cdot \nabla v \, dx + \sum_i \int_{\Gamma_R^i} r_i u v \, ds, \quad (26)$$

$$L(v) = \int_{\Omega} f v \, dx - \sum_i \int_{\Gamma_N^i} g_i v \, ds + \sum_i \int_{\Gamma_R^i} r_i s_i v \, ds. \quad (27)$$

The variational problem then reads: Find u in the appropriate trial space such that

$$a(u, v) = L(v) \quad \forall v \in V. \quad (28)$$

APPENDIX B: THE NON-DIMENSIONALISATION OF HEAT EQUATION

Given the Dirichlet boundaries at $T = 1173$ K and the ambient temperature $T_{\infty} = 323$ K, we choose

$$\Delta T = 1173 - 323 = 850 \text{ K.}$$

Define the dimensionless temperature:

$$\theta = \frac{T - T_{\infty}}{\Delta T}.$$

Thus, at $T = 1173$ K we have $\theta = 1$ and at $T = 323$ K we have $\theta = 0$.

The rectangular domain is given by:

$$0 \leq x \leq L, \quad 0 \leq y \leq H.$$

For the steady-state problem ($\nabla^2 T = 0$):

$$\frac{1}{L^2} \frac{\partial^2 T}{\partial \hat{x}^2} + \frac{1}{H^2} \frac{\partial^2 T}{\partial \hat{y}^2} = 0,$$

or, after multiplying by L^2 :

$$\frac{\partial^2 \theta}{\partial \hat{x}^2} + \left(\frac{L}{H} \right)^2 \frac{\partial^2 \theta}{\partial \hat{y}^2} = 0.$$

At $\hat{x} = 0$ and $\hat{x} = 1$, $\theta = 1$. At $\hat{y} = 0$, $\frac{\partial \theta}{\partial \hat{y}} = 0$. At the top boundary, the dimensional boundary condition is

$$-k \frac{\partial T}{\partial y} = h(T - T_{\infty}) + \epsilon \sigma (T^4 - T_{\infty}^4).$$

Linearize the radiative term about T_{∞} :

$$T^4 - T_{\infty}^4 \approx 4T_{\infty}^3(T - T_{\infty}),$$

so that

$$-k \frac{\partial T}{\partial y} = (h + 4\epsilon\sigma T_{\infty}^3)(T - T_{\infty}).$$

With $T - T_{\infty} = \Delta T \theta$ and $y = H \hat{y}$, we have

$$-k \frac{850}{H} \frac{\partial \theta}{\partial \hat{y}} = (h + 4\epsilon\sigma T_{\infty}^3)(850 \theta).$$

Define the effective Biot number:

$$\text{Bi}_{\text{eff}} = \frac{(h + 4\epsilon\sigma T_{\infty}^3)H}{k}.$$

Thus, the non-dimensional top boundary condition is:

$$-\frac{\partial \theta}{\partial \hat{y}} = \text{Bi}_{\text{eff}} \theta \quad \text{at } \hat{y} = 1.$$

So, the non-dimensionalised heat equation with radiative and convective boundary conditions at the top is:

$$\frac{\partial^2 \theta}{\partial \hat{x}^2} + \left(\frac{L}{H}\right)^2 \frac{\partial^2 \theta}{\partial \hat{y}^2} = 0 \text{ in } \Omega \quad (29)$$

$$\theta = 1 \text{ on } \Gamma_{\text{left/right}} \quad (30)$$

$$\frac{\partial \theta}{\partial \hat{y}} = 0 \text{ on } \Gamma_{\text{bottom}} \quad (31)$$

$$\frac{\partial \theta}{\partial \hat{y}} = \text{Bi}_{\text{eff}} \text{ on } \Gamma_{\text{top}} \quad (32)$$

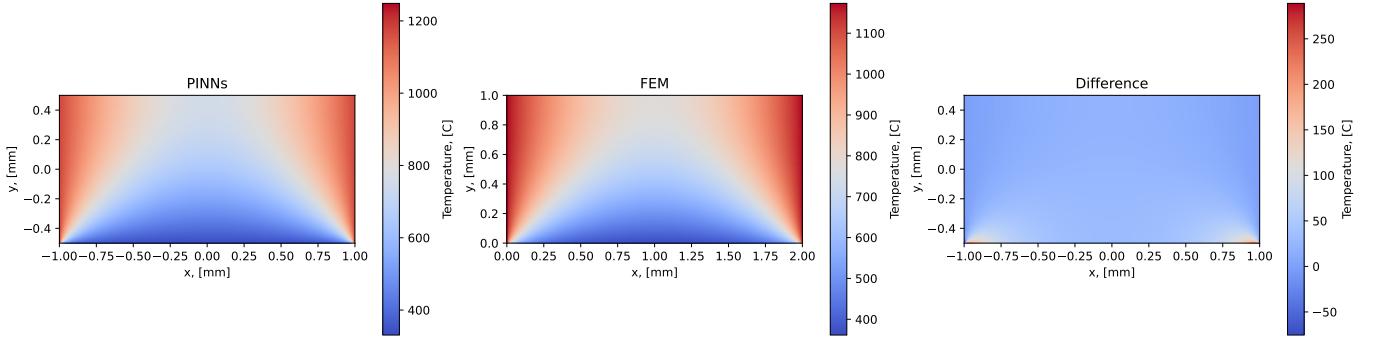


FIG. 15. Spatial comparison of the PINN temperature prediction and FEM reference on the computational domain. Left: PINN-predicted field T_{PINN} . Centre: Reference temperature field T_{ref} . Right: Pointwise absolute error $|T_{\text{PINN}} - T_{\text{ref}}|$

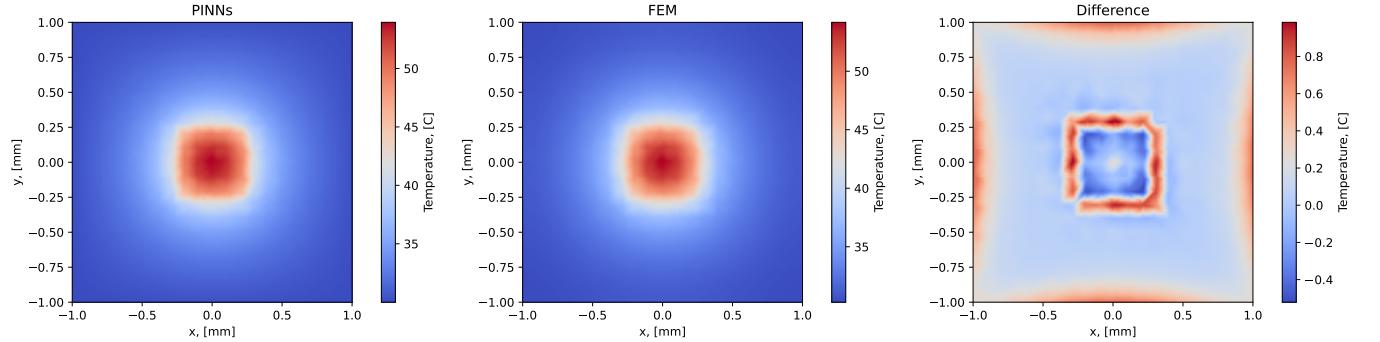


FIG. 16. ($z = -1$) comparison: **left** PINN prediction, **centre** FEM reference, **right** point-wise difference ($\text{PINN} - \text{FEM}$).

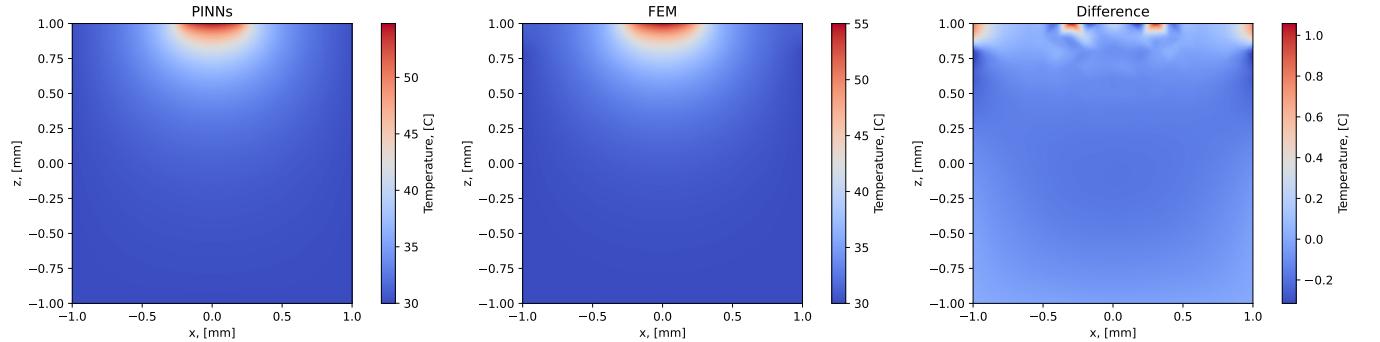


FIG. 17. $y = 0$ slice comparison: **left** PINN prediction, **centre** FEM reference, **right** point-wise difference.

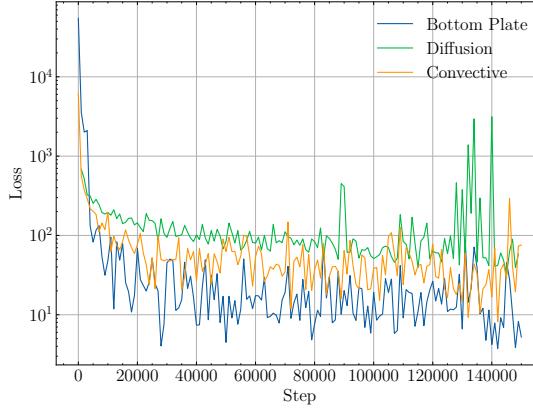


FIG. 18. Evolution of diffusion, convective, and insulated loss components for the fully connected network.

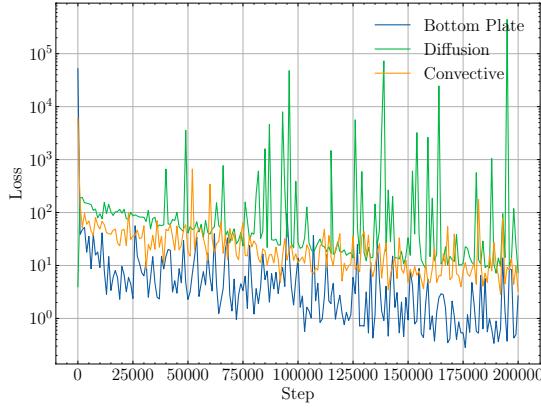


FIG. 19. Evolution of diffusion, convective, and insulated loss components for the Fourier network.

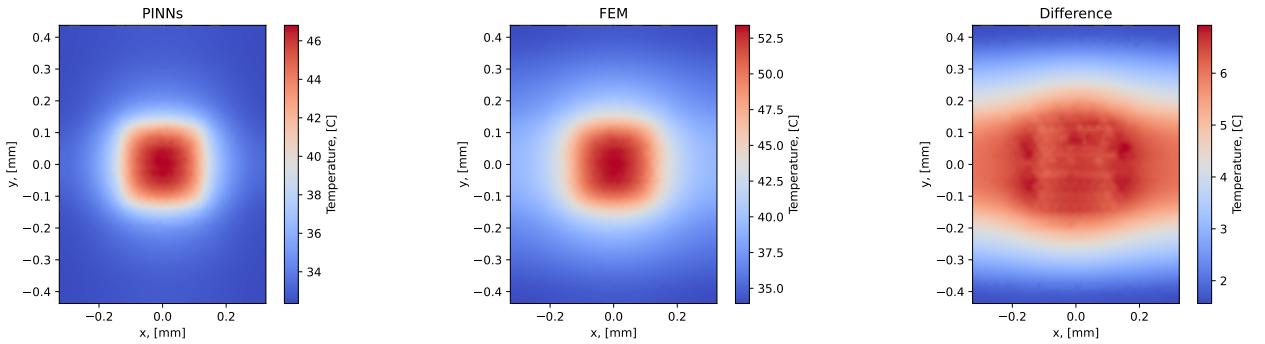


FIG. 20. Slice comparison in the xy -plane for the fully connected network: (left) PINN prediction, (center) FEM reference, (right) pointwise absolute error. The maximal error is above 6 $^{\circ}\text{C}$.

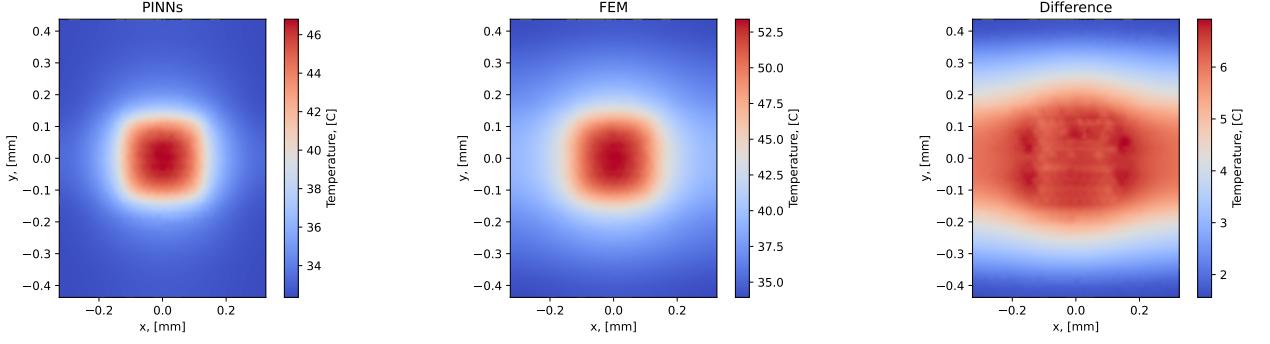


FIG. 21. Slice comparison in the xz -plane for the fully connected network: (left) PINN prediction, (center) FEM reference, (right) pointwise absolute error. The maximal error is above $6\text{ }^{\circ}\text{C}$.

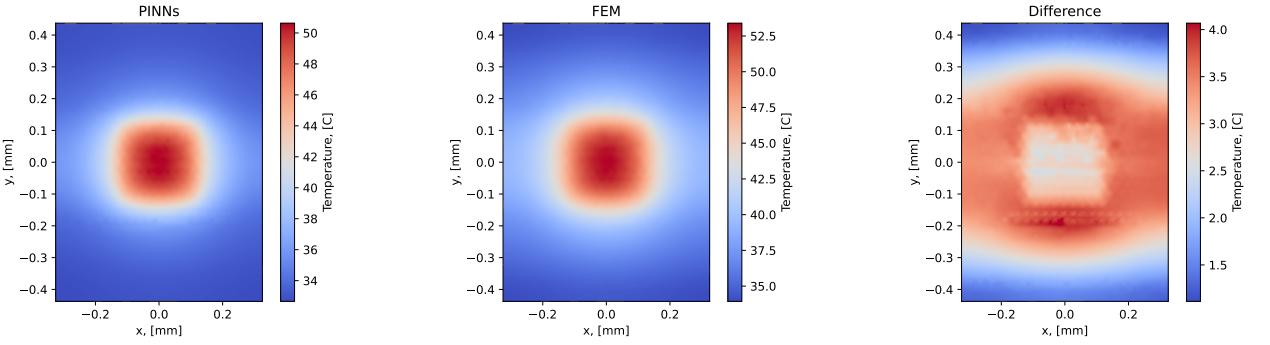


FIG. 22. Slice comparison in the xy -plane for the Fourier network: (left) PINN prediction, (center) FEM reference, (right) pointwise absolute error. The maximal error is $4\text{ }^{\circ}\text{C}$.

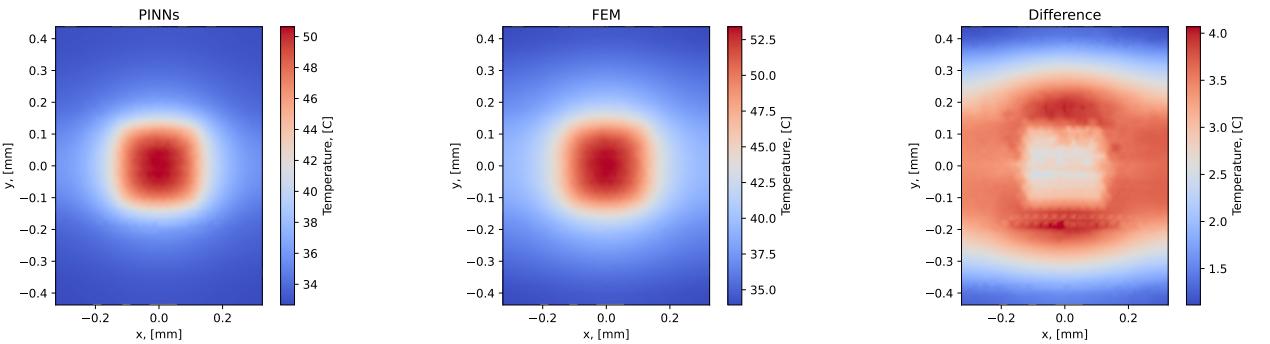


FIG. 23. Slice comparison in the xz -plane for the Fourier network: (left) PINN prediction, (center) FEM reference, (right) pointwise absolute error. The maximal error is $4\text{ }^{\circ}\text{C}$.

-
- [1] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [2] Navid Zobeiry and Keith D. Humfeld. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Engineering Applications of Artificial Intelligence*, 101:104232, 2021.
- [3] Ling Shen, Zhipeng Chen, Xinyi Wang, and Jianjun He. Soft sensor modeling for 3d transient temperature field of large-scale aluminum alloy workpieces based on multi-loss consistency optimization pinn. *Sensors*, 23(14), 2023.
- [4] F. P. Incropera and D. P. DeWitt. *Fundamentals of Heat and Mass Transfer*, 6th ed. JOHN WILEY SONS, 2015.
- [5] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamayr Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations, 2023.
- [6] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains, 2020.
- [7] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [8] PhysicsNeMo Contributors. Nvidia physicsnemo: An open-source framework for physics-based deep learning in science and engineering. <https://github.com/NVIDIA/physicsnemo>, February 2023. Release date: 2023-02-24.
- [9] Igor A. Baratta, Joseph P. Dean, Jørgen S. Dokken, Michal Habera, Jack S. Hale, Chris N. Richardson, Marie E. Rognes, Matthew W. Scroggs, Nathan Sime, and Garth N. Wells. DOLFINx: the next generation FEniCS problem solving environment. preprint, 2023.
- [10] J.P. Holman. *Heat Transfer: Tenth Edition*. McGraw-Hill Education.
- [11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.