Advance Heat Transfer Project Report: Solving 2 heat transfer problems based on PINNs

Hongze Song[1]

[1]China-UK Low Carbon College, Shanghai Jiao Tong University
Course: Advanced Heat Transfer (PE6201) - **Professor:** *Dr. Tao Ren*

Abstract

Heat transfer is a critical topic in engineering and science, requiring accurate solutions to partial differential equations. This project investigates the application of Physics-Informed Neural Networks (PINNs) to solve two-dimensional steady-state heat conduction problems. By extending the PINN framework from a one-dimensional transient case to a two-dimensional domain, I demonstrate its flexibility and adaptability in handling more complex scenarios. The PINN solutions are compared against analytical solutions derived using the method of separation of variables and numerical solutions obtained via the Finite Difference Method (FDM). Through comprehensive error analysis, PINNs are shown to achieve accuracy comparable to traditional methods while incorporating physical laws directly into the learning process. Furthermore, network optimization techniques, including dynamic learning rate adjustment, significantly improved convergence and solution accuracy. This work highlights the effectiveness of PINNs for solving heat transfer problems and establishes a foundation for future research on more complex physical systems.

1 Introduction

1.1 Background

Physics-Informed Neural Networks (PINNs) represent a novel framework that integrates physical laws with deep learning by embedding the constraints of partial differential equations directly into the loss function of the neural network. By leveraging automatic differentiation to accurately handle spatial and temporal derivatives, PINNs eliminate the need for grid discretization and are particularly effective in scenarios with sparse or high-cost data. Compared to traditional numerical methods, PINNs demonstrate remarkable flexibility and predictive capabilities in tackling complex problems such as fluid mechanics and quantum mechanics, opening new avenues for physical modeling and system identification.

1.2 literature review

Raissi et al.[**1**] proposed a novel computational framework, Physics-Informed Neural Networks (PINNs), which integrates deep learning with physical laws for solving and discovering partial differential equations (PDEs). By embedding the physical constraints of PDEs directly into the loss function of neural networks, PINNs ensure that the solution not only approximates the unknown function but also adheres to the governing physical laws. In the Data-driven solutions of partial differential equations section, PINNs introduce two main approaches: Continuous Time Models and Discrete Time Models.

In the Continuous Time Models, PDEs are treated as continuous functions of time and space. Neural networks are used to approximate the solution $u_\theta(t,x)$, while automatic differentiation is employed to compute temporal and spatial derivatives, which are used to enforce the residual of the PDE in the loss function. For example, for a general PDE:

$$u_t + \mathcal{N}[u] = 0, \quad x \in \Omega,\ t \in [0,T], \tag{1}$$

where $u(t,x)$ denotes the latent (hidden) solution, $\mathcal{N}[\cdot]$ is a nonlinear differential operator, and Ω is a subset of $\mathbb{R}^D$.

With the definition of the LHS equation; i.e. $f := u_t + \mathcal{N}[u]$, the shared parameters between the neural networks $u(t,x)$ and f(t,x) can be learned by minimizing the mean squared error loss

$$MSE = MSE_u + MSE_f, \tag{2}$$

where

$$MSE_u = \frac{1}{N_u}\sum_{i=1}^{N_u}\left|u(t_u^i, x_u^i) - u^i\right|^2,$$

and

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f}\left|f(t_f^i, x_f^i)\right|^2.$$

MSE_u represents the error in fitting initial and boundary conditions, and MSE_f minimizes the residual $f_\theta(t,x)$. By optimizing this loss function, the neural network is trained to approximate the solution while satisfying the physical constraints of the PDE.

In the Discrete Time Models, PDEs are discretized into a series of time steps, with numerical methods such as the Runge-Kutta method employed to propagate the solution forward in time. The general Runge-Kutta formulation can be expressed as:

$$u^{n+1} = u^n + \Delta t \sum_{j=1}^{q} b_j \mathcal{N}[u^{n+c_j}]$$

where b_j and c_j are numerical coefficients. To compute the solution at each time step, the solution u^{n+1} is further decomposed into multiple intermediate stages:

$$u^{n+c_i} = u^n + \Delta t \sum_{j=1}^{q} a_{ij} \mathcal{N}[u^{n+c_j}], \quad i = 1, \dots, q$$

where a_{ij} are coefficients of the Runge-Kutta scheme. The neural network predicts $[u^{n+c_1}, \dots, u^{n+c_q}, u^{n+1}]$ and minimizes the residuals of these stages to approximate the time evolution of the solution.

Furthermore, in the *Data-driven discovery of partial differential equations* section, PINNs extend their application to inverse problems, aiming to discover hidden physical laws or PDEs from limited observational data. Assuming an unknown PDE in the form:

$$\mathcal{N}[u; \lambda] = 0$$

where λ represents the unknown parameters, the network optimizes the loss function:

$$MSE = MSE_u + MSE_\lambda$$

to simultaneously fit the data and identify the structure and parameters of the PDE.

This study emphasizes the general methodologies of PINNs in both Continuous Time Models and Discrete Time Models, demonstrating their versatility in solving and discovering PDEs by embedding neural networks with physical constraints.

2 Problem setup

In this project, I begin by addressing the one-dimensional transient heat conduction problem as a foundation for understanding the applicability of Physics-Informed Neural Networks (PINNs) to solving partial differential equations. Through this exploration,I identify that the key distinction between one-dimensional transient problems and two-dimensional steady-state problems lies in the nature of the input variables to the neural network. By replacing the temporal input in the transient case with a spatial dimension, the PINN framework is extended to handle the more complex two-dimensional steady-state heat conduction problem. Subsequently, I focus on the two-dimensional problem, conducting a thorough investigation that includes the construction of the neural network, comparison with analytical and numerical solutions, and the optimization of the network architecture to improve accuracy and efficiency.

2.1 1D-trasient Heat Transfer

The governing equation for the one-dimensional transient heat conduction problem is given by:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{3}$$

where $u(x,t)$ represents the temperature distribution at position x and time t, and α denotes the thermal diffusivity of the material.

To fully define the problem, the following initial and boundary conditions are imposed:

- **Initial Condition:**

  $$u(x, 0) = \sin(\pi x),$$

  indicating that the initial temperature distribution follows a sine function.
- **Boundary Conditions:**

  $$u(-1, t) = 0, \quad u(1, t) = 0$$

  corresponding to fixed-temperature boundary conditions where the temperature at both ends is maintained at zero.

The objective of this problem is to determine the temperature distribution $u(x,t)$ that satisfies the governing equation along with the specified initial and boundary conditions, thereby characterizing the temporal and spatial evolution of the heat conduction process.

2.2 2D-steady Heat Transfer

In this project, I focus on the modeling and solution of a two-dimensional steady-state heat conduction problem.

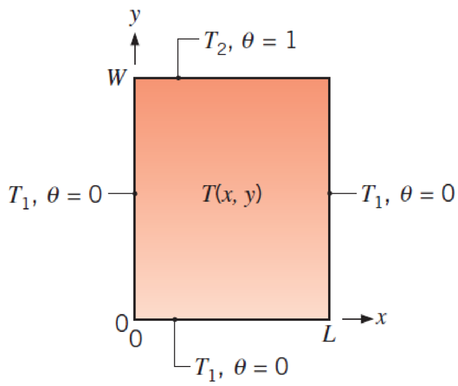

Figure 1: *2D-steady heat transfer problem setup*

The governing equation for this problem is given by:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \tag{4}$$

where $T(x,y)$ represents the temperature distribution within the two-dimensional domain under steady-state conditions with no heat sources.

The boundary conditions:

- The left, bottom, and right boundaries are maintained at a constant temperature T_1,
- The top boundary is maintained at a temperature T_2.

which can be expressed as:

$$\begin{cases} T(0,y) = T_1, & T(L,y) = T_1, \\ T(x,0) = T_1, & T(x,W) = T_2 \end{cases} \tag{5}$$

where L and W denote the length and width of the rectangle, respectively.

To simplify the problem and introduce a dimensionless variable, we define the non-dimensionalized temperature θ as:

$$\theta = \frac{T - T_1}{T_2 - T_1}.$$

With this transformation, the governing equation and boundary conditions are as follows:

$$\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} = 0, \tag{6}$$

$$\begin{cases} \theta(0,y) = 0, & \theta(L,y) = 0, \\ \theta(x,0) = 0, & \theta(x,W) = 1. \end{cases} \tag{7}$$

3 Solutions

3.1 Construction of PINN

In this project, a unified neural network architecture is employed to solve both and two-dimensional steady-state problems, as illustrated in Figure 2. The neural network consists of an 2×2 input layer, 6 multiple hidden layers, and an output layer. The input layer takes spatial coordinates (x, y) as inputs (with the inclusion of time t for transient problems). The hidden layers process the input through nonlinear activation functions *Tanh* to extract features, and the output layer predicts the corresponding temperature values . Leveraging automatic differentiation, the network is capable of computing second-order spatial derivatives and first-order temporal derivatives of the output, enabling it to enforce the governing physical equations.

The training dataset comprises two types of points as figure3: boundary points and interior points. Boundary points, evenly distributed along each edge, are used to impose boundary conditions, with a total of 100 points per boundary. Interior points, randomly sampled across the domain, are used to enforce the residuals of the governing equations, with a total of 6000 points. The neural network maps these points to the predicted temperature distribution while minimizing the residuals of both the boundary conditions 7 and the governing equations 6 during training. This unified architecture is not only suitable for solving two-dimensional steady-state problems but can also for time-dependent one-dimensional transient problems.

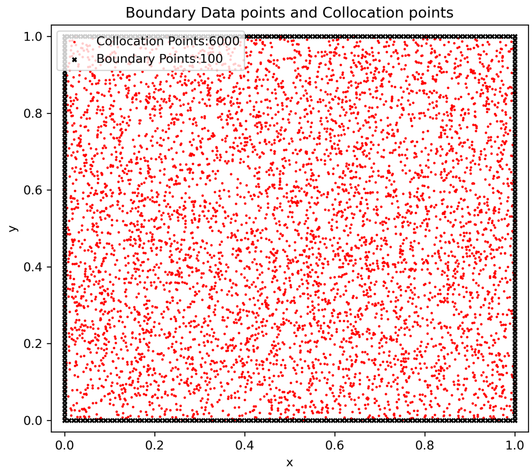

Figure 3: *2D-steady heat transfer problem setup*

Physics Loss

The mathematical formulation of the physics loss is:

$$\mathcal{L}_{\text{physics}} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2}\right)^2 \tag{8}$$

where N denotes the number of sampled collocation points, and $\frac{\partial^2 \theta}{\partial x^2}$ and $\frac{\partial^2 \theta}{\partial y^2}$ represent the second-order derivatives of the network output $\theta(x,y)$ with respect to x and y, respectively. By minimizing $\mathcal{L}_{\text{physics}}$, the model learns solutions that satisfy the governing equation.

Boundary Loss

The boundary loss is defined as:

$$\mathcal{L}_{\text{boundary}} = \frac{1}{N_y}\sum_{i=1}^{N_y}\left[\theta(0,y_i)^2 + \theta(L,y_i)^2\right] + \frac{1}{N_x}\sum_{i=1}^{N_x}\left[\theta(x_i,0)^2 + (\theta(x_i,W)-1)^2\right] \tag{9}$$

where N_y and N_x are the numbers of boundary points sampled along the y-direction and x-

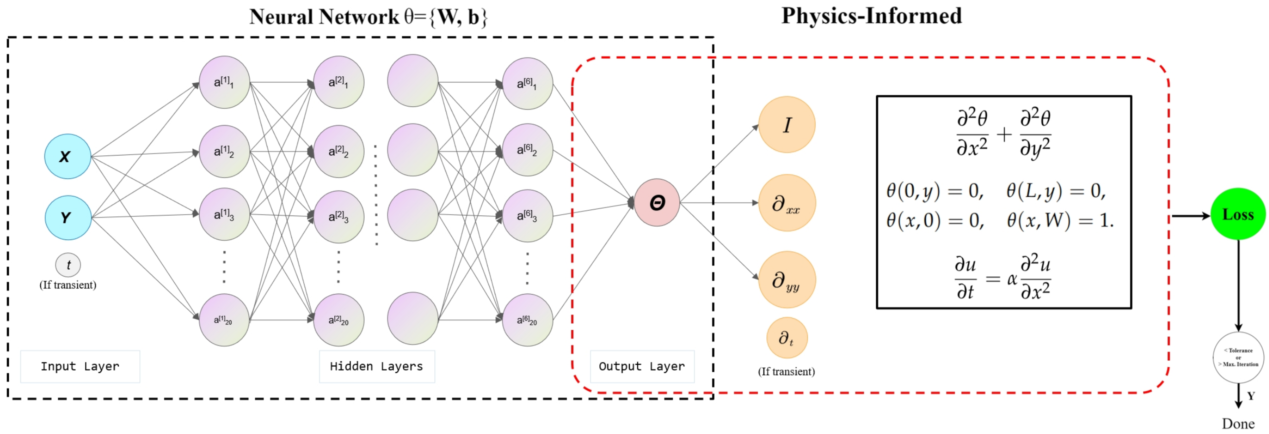

Figure 2: *Construction of neural network*

direction, respectively. Minimizing $\mathcal{L}_{\text{boundary}}$ ensures that the predicted temperature values satisfy the given boundary conditions.

Total Loss Function

The total loss function for training the PINN model is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{physics}} + \mathcal{L}_{\text{boundary}}. \tag{10}$$

3.2 Ground Truth Solutions

In this project, to analyze the error of the neural network predictions, I employed analytical and numerical methods to obtain ground truth solutions. For the one-dimensional transient problem, although a complete numerical solution using the Finite Difference Method (FDM) was not performed, the accuracy of the neural network predictions was evaluated by verifying their consistency with the true values at boundary points.

For the two-dimensional steady-state problem, both analytical and numerical solutions were used as reference standards.

Analytical Solution

To directly use the method of separation of variables, assuming the solution can be expressed as the product of spatially dependent variables:

$$\theta(x,y) = X(x)Y(y),$$

where $X(x)$ and $Y(y)$ are functions solely dependent on x and y, respectively.

By incorporating the boundary conditions, the analytical solution is expressed as:

$$\theta(x,y) = \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^{n+1}+1}{n}\sin\left(\frac{n\pi x}{L}\right)\frac{\sinh\left(\frac{n\pi y}{L}\right)}{\sinh\left(\frac{n\pi W}{L}\right)} \tag{11}$$

Numerical Solution

Additionally, the Finite Difference Method (FDM) was employed to solve the two-dimensional problem numerically. By discretizing the governing equation:

$$\theta_{i,j} = \frac{\theta_{i-1,j} + \theta_{i+1,j} + \theta_{i,j+1} + \theta_{i,j-1}}{4} \tag{13}$$

The two-dimensional domain was discretized into a grid of $(Nx+1)\times(Ny+1) = 100\times 100$ points, and iterative methods were used to solve the resulting system of equations, yielding the temperature distribution.

4 Results

In this section, we present the PINN solutions for both the one-dimensional transient problem and the two-dimensional steady-state problem, along with comparisons to analytical and numerical solutions. By visualizing the results, analyzing error distributions, and conducting quantitative evaluations, the accuracy and applicability of PINNs in solving heat transfer problems are comprehensively validated.

4.1 One-Dimensional Transient Problem

For the one-dimensional transient heat transfer problem, Figure 4.c and 4.d illustrates the temperature distribution obtained using the PINN approach, showing the temperature evolution over the spatial domain at various time steps. Additionally, the loss convergence curve during training, shown in Figure 4.a, demonstrates that the model efficiently converges to a stable solution, with the final loss value reaching $loss = 0.0002458$. This low loss value indicates that the PINN solution satisfies both the governing equation and the boundary conditions with high accuracy.

$$kdy\frac{T_{i-1,j}-T_{i,j}}{\Delta x}+kdy\frac{T_{i+1,j}-T_{i,j}}{\Delta x}+kdx\frac{T_{i,j+1}-T_{i,j}}{\Delta y}+kdx\frac{T_{i,j-1}-T_{i,j}}{\Delta y}=0 \tag{12}$$

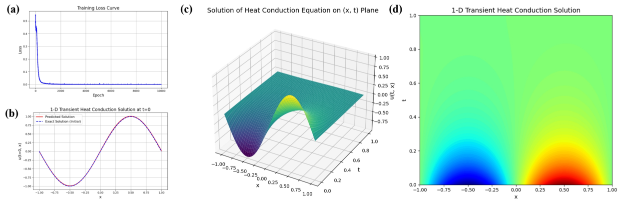

Figure 4: *1D-transient heat transfer solution*

To evaluate the accuracy of the predictions, Figure 4.b shows the predicted temperature values at the initial $t = 0$ conditions to the true values. The results show perfect agreement, confirming that the PINN model accurately captures the physical laws governing the one-dimensional transient problem.

4.2 Two-Dimensional Steady-State Problem

For the two-dimensional steady-state heat transfer problem, Figures 5 present the 2D and 3D visualizations of the temperature distribution obtained using the PINN approach. The training loss curve, shown in Figure 6, initially displayed significant oscillations when using the same learning rate as the one-dimensional transient problem. To address this issue, a dynamic learning rate adjustment strategy was implemented, which effectively stabilized the training process and improved the convergence behavior. And the final loss value reaches $loss = 0.0264$

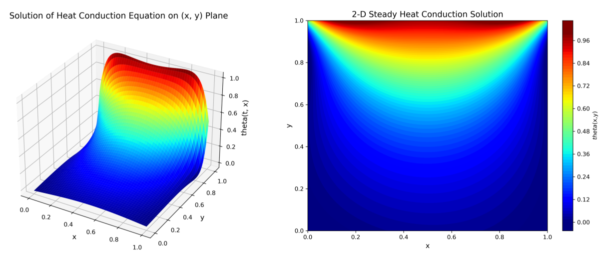

Figure 5: *2D-steady heat transfer solution*

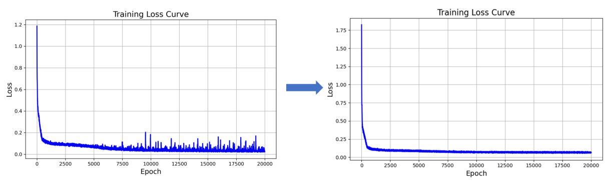

Figure 6: *Loss Curve before and after optimization. The left figure uses a fixed learning rate, while the right figure using a dynamically adjusted learning rate that decreases with training epochs.*

PINN shows good overall performance in predicting 2D steady-state heat conduction. The predicted temperature distribution matches the true solution well, especially in the central region. However, noticeable errors appear near the boundaries, indicating challenges in handling boundary conditions.

4.3 Error Analysis

To quantify the accuracy of the PINN predictions, we compared them to the numerical solution obtained via the Finite Difference Method (FDM) and the analytical solution. From a global perspective, the results from all three methods show high consistency as figure 9. Furthermore, global error analysis (Figure 10) reveals that the global error of the PINN solution relative to the analytical solution is within an acceptable range and is comparable to that of the numerical method.

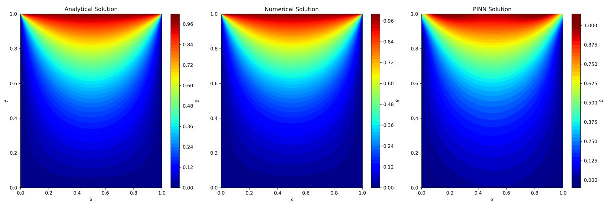

Figure 7: *Results from all three methods*

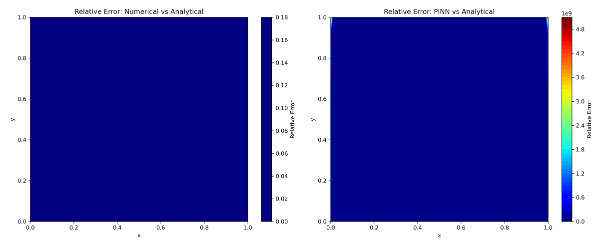

Figure 8: *Comparison of Relative Mean Squared Error (MSE) Between Numerical, PINN, and Analytical Solutions*

Table 1: *MSE of PINN and numerical solution*

	Numerical	PINN
MSE	9.94e-05	2.89e-04

Table 1 summarizes the global mean squared errors (MSE) of the PINN solution and the numerical solution relative to the analytical solution, further confirming the high accuracy of the PINN approach in solving the two-dimensional steady-state problem.

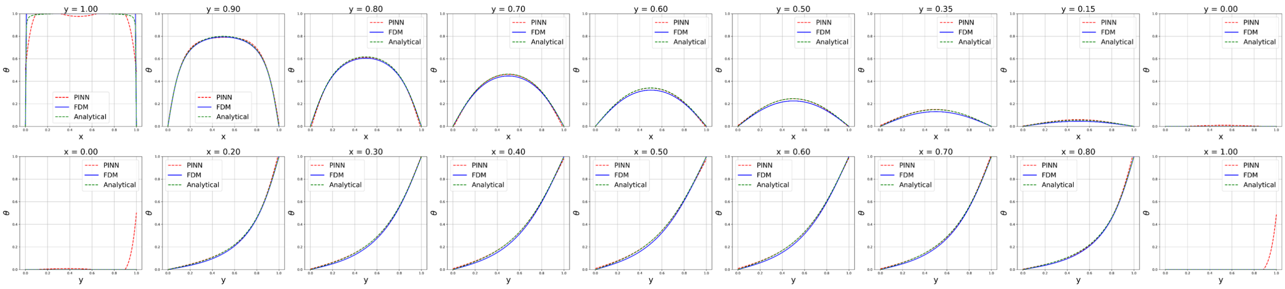

Figure 9: *Comparison of Local Temperature Profiles*

Additionally, to perform a more detailed comparison of the solutions from different methods, I fixed specific x or y coordinates and compared the local temperature distributions (Figure 11). The results demonstrate that the PINN solution aligns closely with both the numerical solution and the analytical solution across the entire domain.

Closer inspection of the error distribution shows that larger errors are concentrated near the boundaries (upper left corner and upper right corner), where enforcing boundary conditions presents a greater challenge. However, for internal points, the predicted temperature values closely match the true values.

4.3 Summary

In summary, the results of both the one-dimensional transient and two-dimensional steady-state problems demonstrate the significant advantages of the PINN approach in solving heat transfer problems. The model not only accurately predicts temperature distributions but also exhibits high consistency in boundary condition validation. Through comparisons with numerical and analytical solutions, the global and local accuracy of the PINN predictions has been thoroughly verified. Although larger errors are observed near the boundary regions, the temperature distribution within the interior closely matches the true values, highlighting the potential of PINNs in addressing complex physical problems. Furthermore, with the implementation of dynamic optimization strategies during training, the PINN model achieves improved convergence performance and solution accuracy.

5 Discussion

While the results demonstrate the effectiveness of Physics-Informed Neural Networks (PINNs) in solving heat transfer problems, there remain several areas where the method can be further improved to enhance its performance and robustness.

Neural Network Optimization

The architecture and hyperparameters of the neural network play a crucial role in determining the accuracy and convergence speed of PINNs. In this study, a simple feedforward neural network was utilized, but more advanced architectures, such as deep residual networks (ResNets) or adaptive activation functions, could be explored to improve the model's expressive capability. Additionally, optimization techniques such as batch normalization, and early stopping may further enhance convergence and prevent overfitting.

Boundary Prediction Accuracy

One of the primary challenges identified in this study is the relatively larger errors observed near the boundary regions. Future work could explore strategies to address this, such as increasing the density of collocation points along boundaries, employing boundary-aware loss weighting schemes, or incorporating additional regularization terms to penalize boundary discrepancies more effectively. Besides, Use adaptive activation functions, such as SIREN[3] (activation with sine function), which is more efficient when dealing with boundary conditions.

References

[1] Raissi M, Perdikaris P, Karniadakis G E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J]. Journal of Computational physics, 2019, 378: 686-707.

[2] CSDN. The principle and implementation of using PINN to solve the heat conduction equation [EB/OL]. CSDN Blog, [2025-01-13]. Available: `https://blog.csdn.net/m0_68330627/article/details/143652512`.

[3] Sitzmann V, Martel J, Bergman A, et al. Implicit neural representations with periodic activation functions[J]. Advances in neural information processing systems, 2020, 33: 7462-7473.