MATLAB EXPO

2024.06.11 | 그랜드 인터컨티넨탈 서울 파르나스

Development of Physics-Based AI Systems: Focusing on Neural Operator and PINN

Hyung Ju Hwang, POSTECH

MathWorks®

Digital Twin in Industry

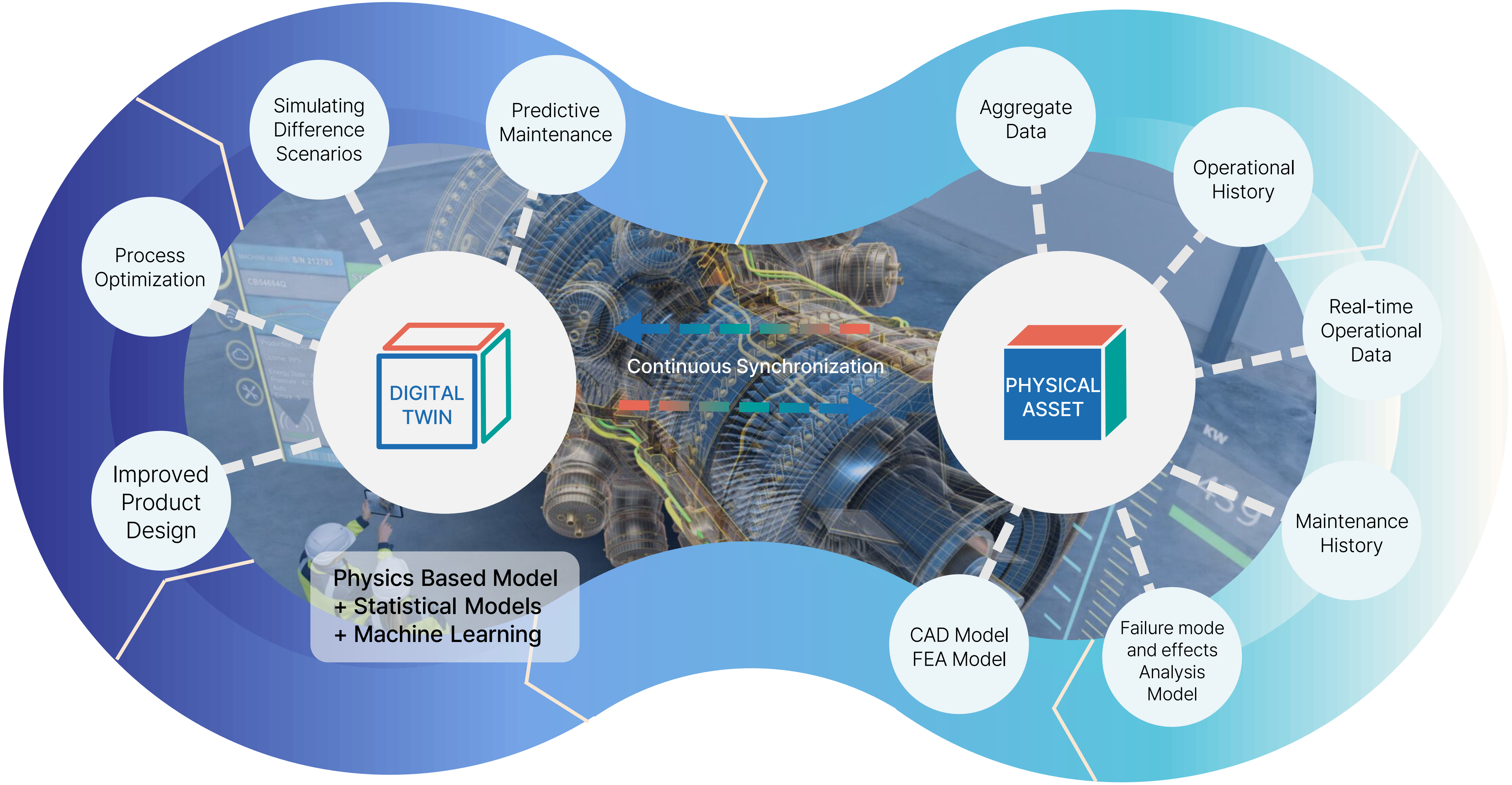


Table of Contents

Part I. Data and Physics Models

PDE Solvers

PDE Solvers

Neural-enhanced Classical Method

Neural Networks assist Numerical methods

- ML based Spectral Method
- Neural Iterator
- Solver-in-the-loop

Classically Inspired Neural Methods

Implementing Numerical Methods using Neural Network

- Message Passing PDE
- PDE-Net
- PDE-Graph Convolution Net

Fully Neural Methods

Universal Approximation based on Neural Networks

Today's Topic

- DeepONet
- Galerkin Transformer
- Fourier Neural Operator

Supervised Learning

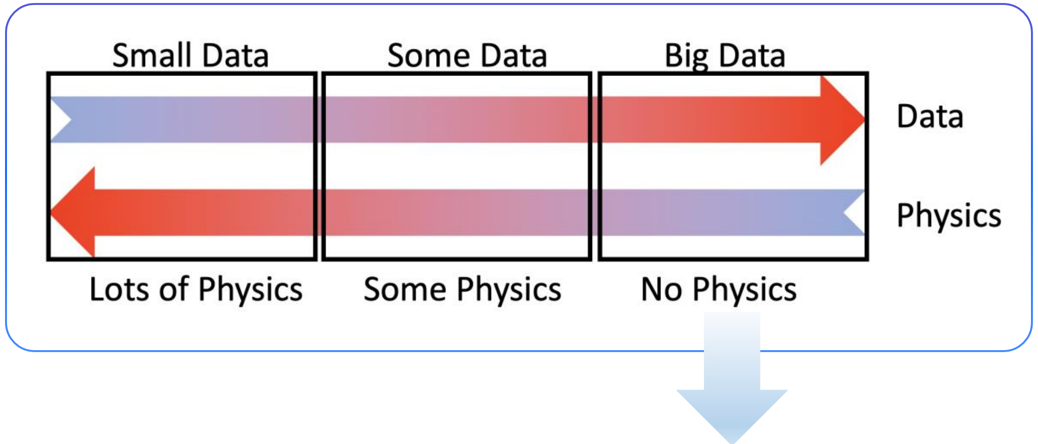
- Physics Informed Neural Network (PINN)
- Physics Informed Machine Learning

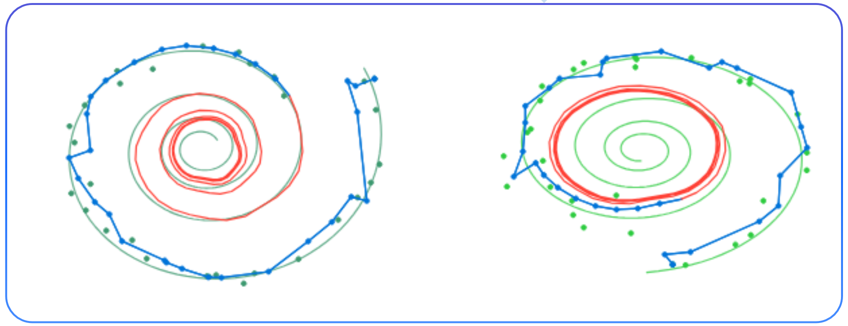
Unsupervised
Semi-supervised

Why research into physics-based AI is necessary

Reason 1.

Pure neural networks are unphysical!

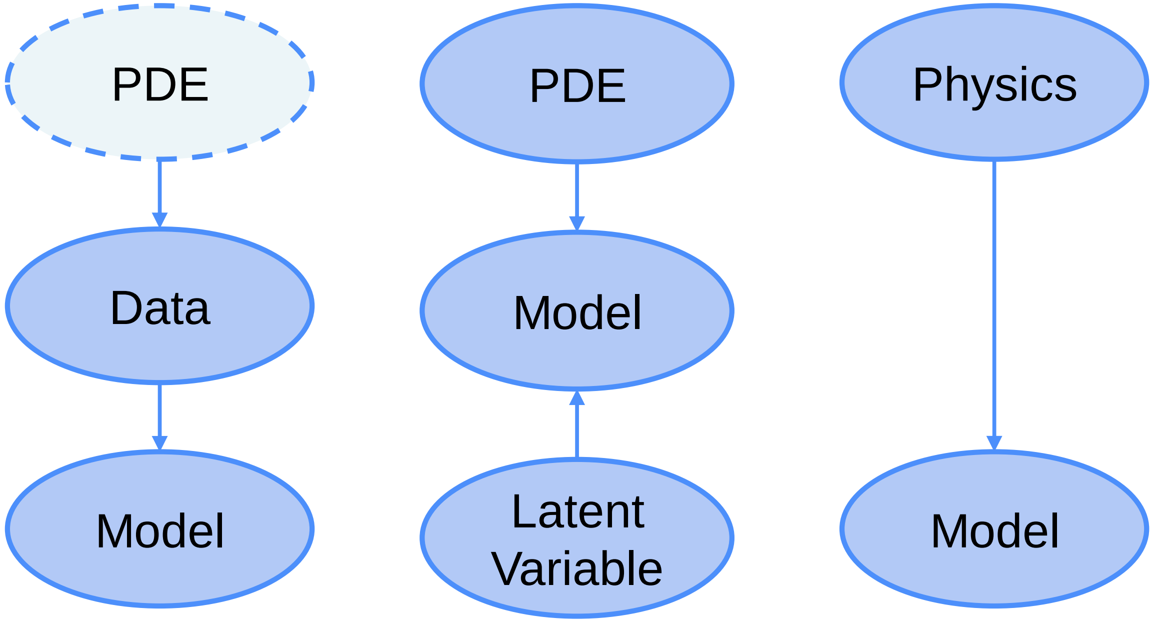




RNN that cannot learn a simple ODE system

Reason 2.

There is no magic solution.

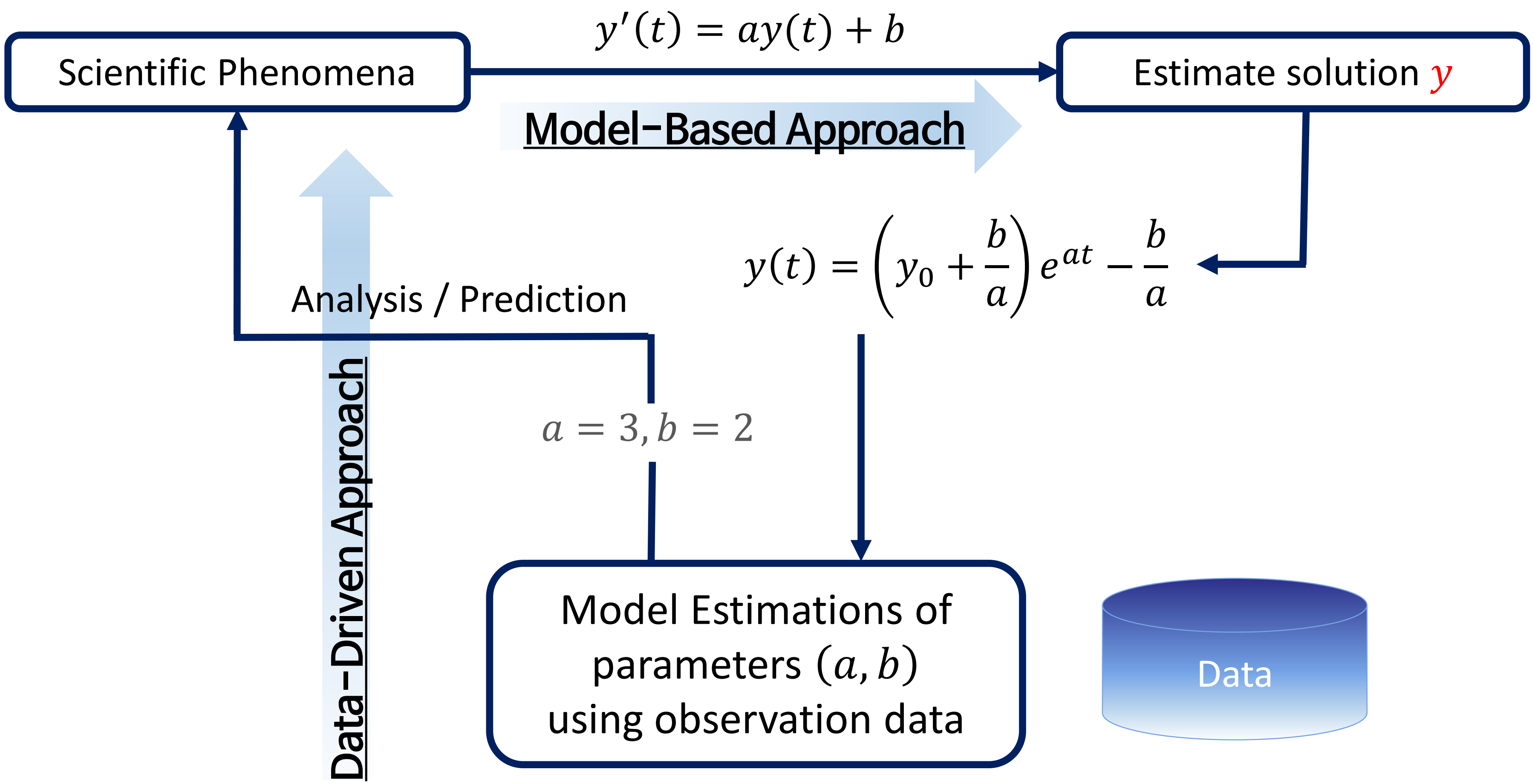


Scenario1.
Only observational data available
Scenario2.
Only numerical analysis data available

Scenario1.
PDE available
but parameters unknown
Scenario2.
PDE available
but poorly modeled

Scenario1.
Only PDE available

Data-driven vs. model-driven vs. data & model



Part II. Neural PDE Solvers

Deep Learning Approach to PDEs

Deep Learning Approach to PDEs : PINN

PDE Solver

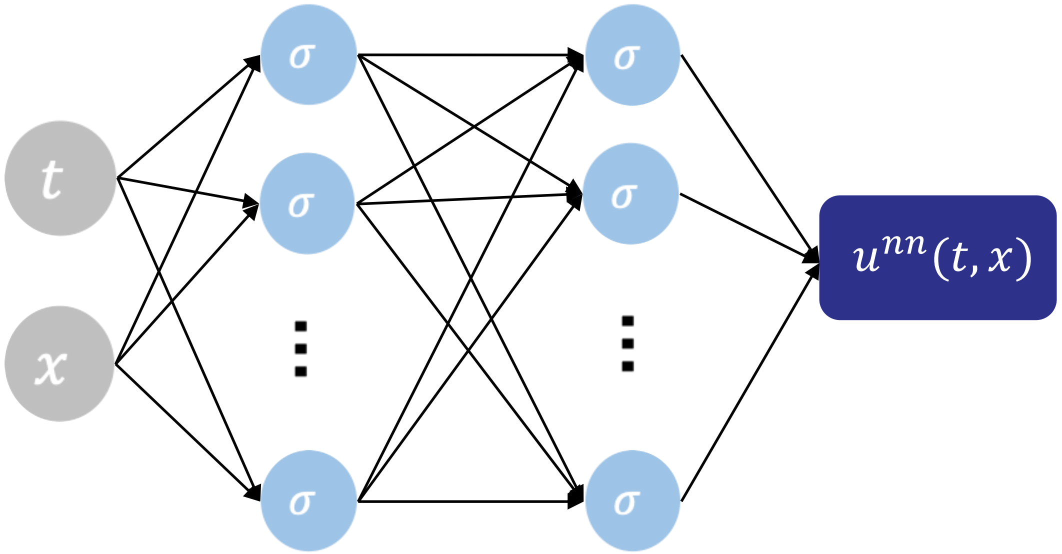
- Using neural networks directly to parametrize the solution to PDEs.
- Solve one instance of PDE at a time.

Find $\boldsymbol{u(t,x)}$ satisfying

$$\mathcal{L}_{PDE} = f(u, u_t, u_x, u_{xx}, \dots) = 0$$

$$\mathcal{L}_{IC} = u(0,x) - g(x) = 0$$

$$\mathcal{L}_{BC} = u|_{\partial\Omega} - h(t,x)|_{\partial\Omega} = 0$$



Deep Learning Approach to PDEs : Operator Learning

Operator Learning

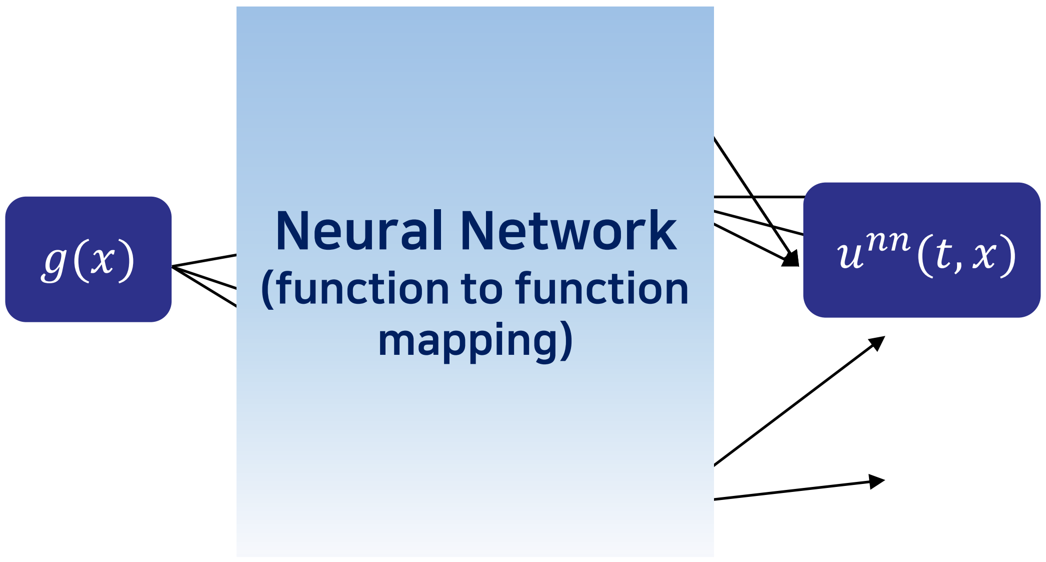
- Learning a mapping from the parameters of the PDEs to the corresponding solution.
- Learning a family of PDEs from data.

Find a map $\mathcal{G}: g(x) \mapsto u(t,x)$ satisfying

$$\mathcal{L}_{PDE} = f(u, u_t, u_x, u_{xx}, \dots) = 0$$

$$\mathcal{L}_{IC} = u(0,x) - g(x) = 0$$

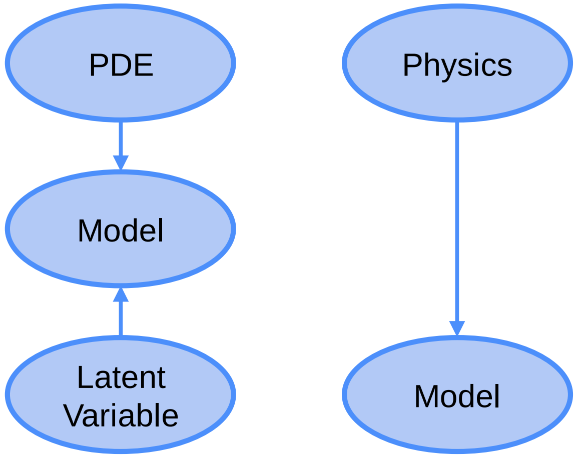
$$\mathcal{L}_{BC} = u|_{\partial\Omega} - h(t,x)|_{\partial\Omega}$$



Part II. Neural PDE Solvers

PINN to Forward-Inverse Problems

This part introduces several strategies to solve the below problem settings



Introduction: What is Forward-Inverse problem?

Forward problem	▪ Find a solution of a given differential equation
Inverse problem	▪ **Estimate model parameters or coefficients of the model(latent variables)** based on the observed data
Forward-Inverse problem	▪ Solve forward and inverse problems based on data

Introduction: Forward-Inverse problems

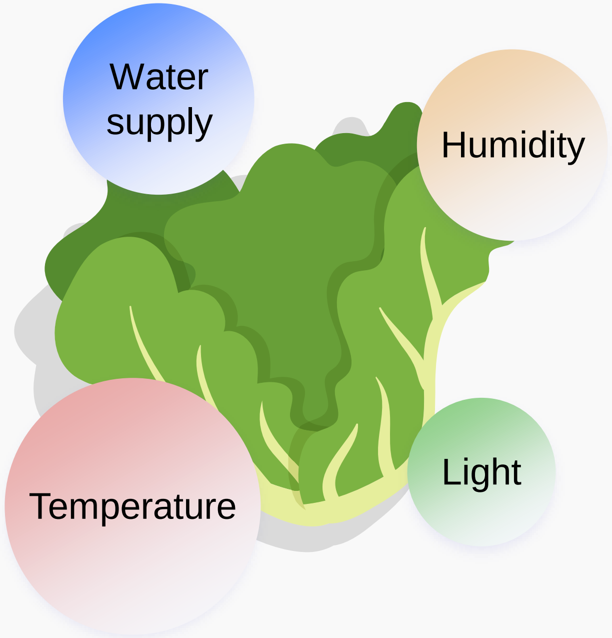
GOALS

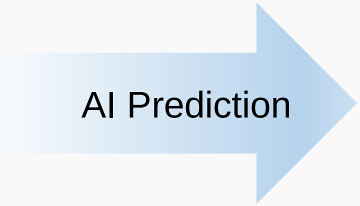
1**. An easier way** to solve differential equations.

2. Reducing **two steps to a single step.**

Predicting the Growth of Lettuce

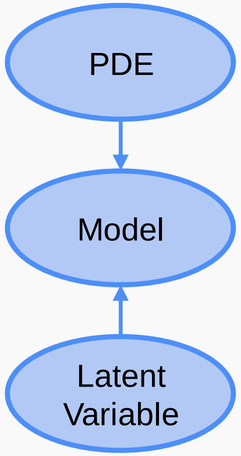
- Goal : to find the optimal growth environment to derive the daily maximum leaf weight of lettuce





Environment variables

Mass of Lettuce



Predicting the Growth of Lettuce

Logistic growth model

$$\frac{dy}{dt} = \mathbf{k}y(1 - \frac{y}{a})$$

However, this model does not reflect the growth conditions of lettuce

Data

- Hourly Humidity, Temperature, and Water Supply
- Hourly Changing Mass

Latent variable
(Humidity, Temperature, and Water Supply)

↓

Neural Network

↓

$k(t)$

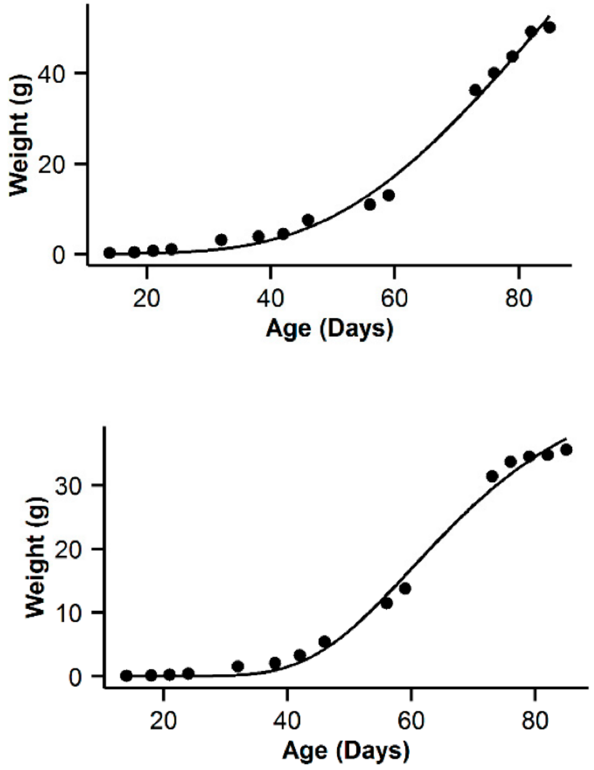
↓

Logistic growth model

↓

Mass of Lattuce

Neural ODE Solver



Possible to Explore Optimal Control Conditions (Humidity, Temperature, Water Supply)

COVID-19 Case Data

Data Used

Status of confirmed cases by region, March 25, 2020.

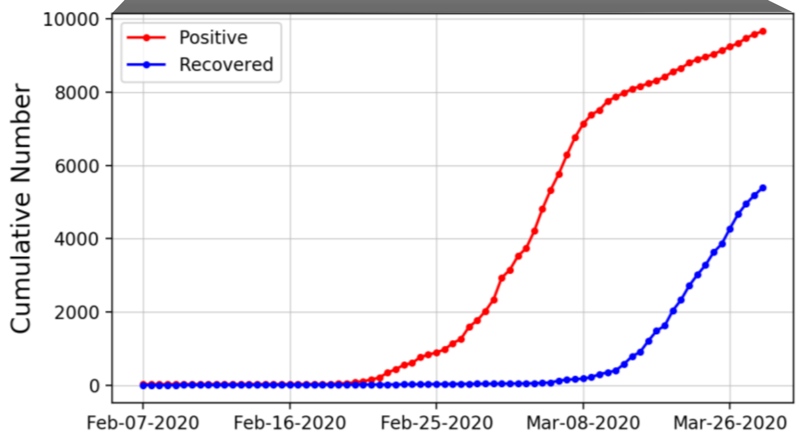
Period: February 7 - March 30, 2020, South Korea

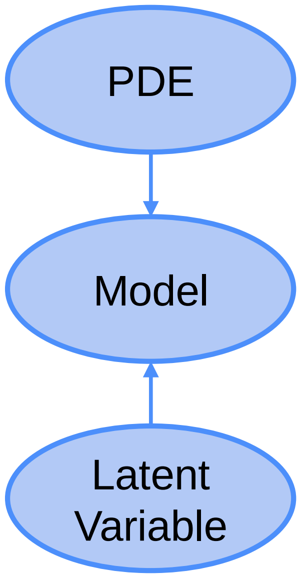
	Total	Seoul	Busan	…	Jeju
Confirmed	681	165	10	…	1
Recovered	10,275	614	131	…	13
Deceased	269	4	3	…	0
Total	11,225	783	144	…	14

Korea Disease Control and Prevention Agency (www.kdca.go.kr)

Cumulative Number of Positive Cases Nationwide in South Korea

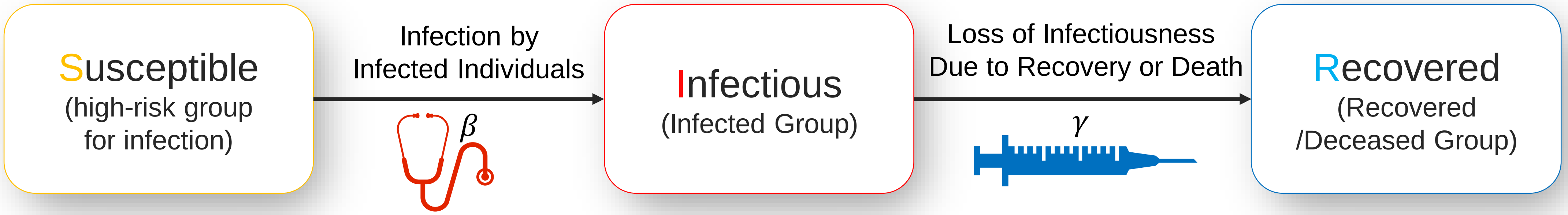
Daily Recovered and Deceased Numbers Nationwide in South Korea





COVID-19 Spread Prediction and Prevention Policies

Mathematical Model



$$\frac{dS}{dt} = -\beta S I$$

$$\frac{dI}{dt} = \beta S I - \gamma I$$

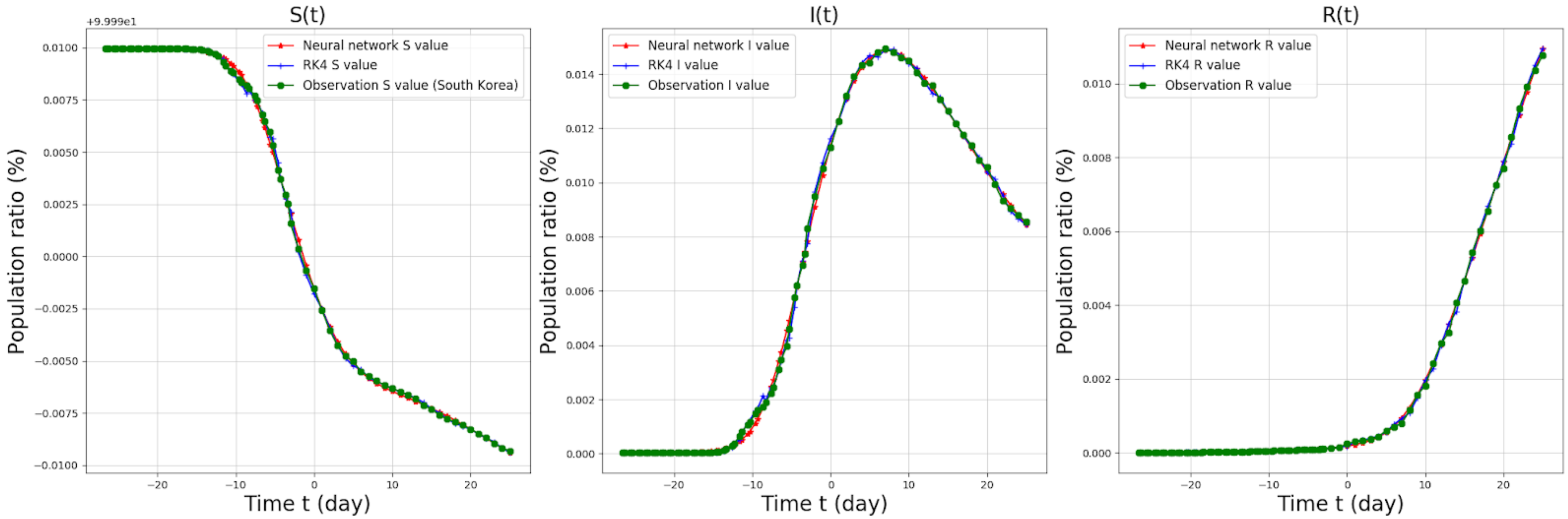
$$\frac{dR}{dt} = \gamma I$$

$$N \coloneqq S + I + R = 1$$

(Considered as a Proportion of the Population)

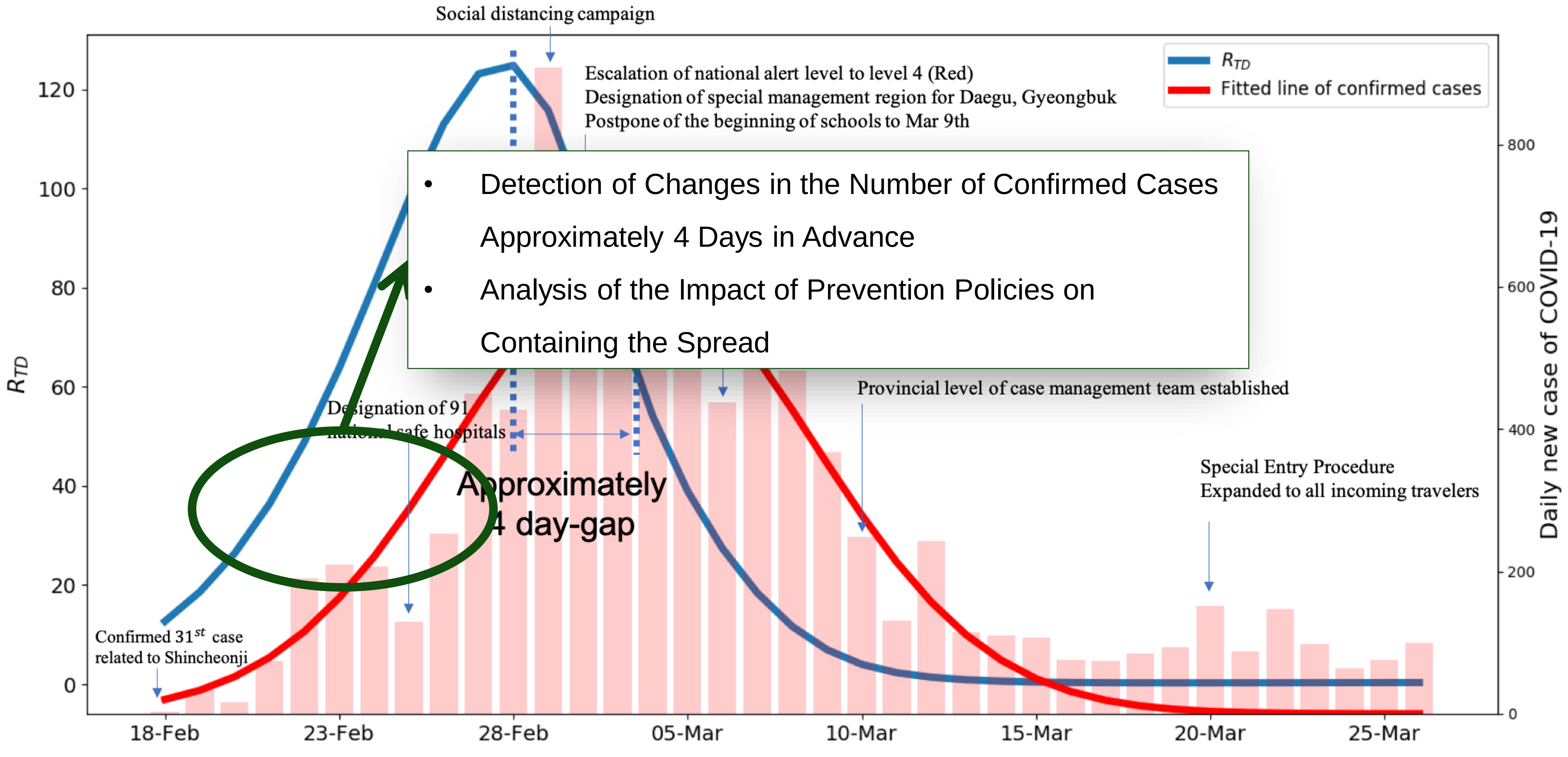
COVID-19 Spread Prediction and Prevention Policies

Change Patterns : Results Using Deep Learning



S(t)	I(t)	R(t)
high-risk group for infection	Infected Group	Recovered/Deceased Group

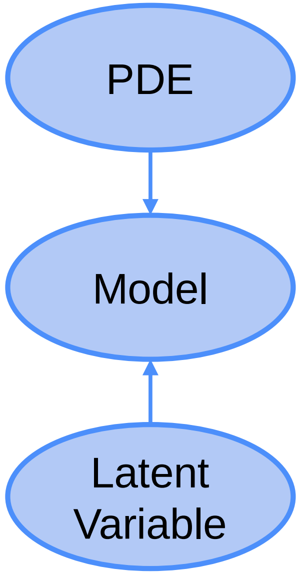
COVID-19 Spread Prediction and Prevention Policies

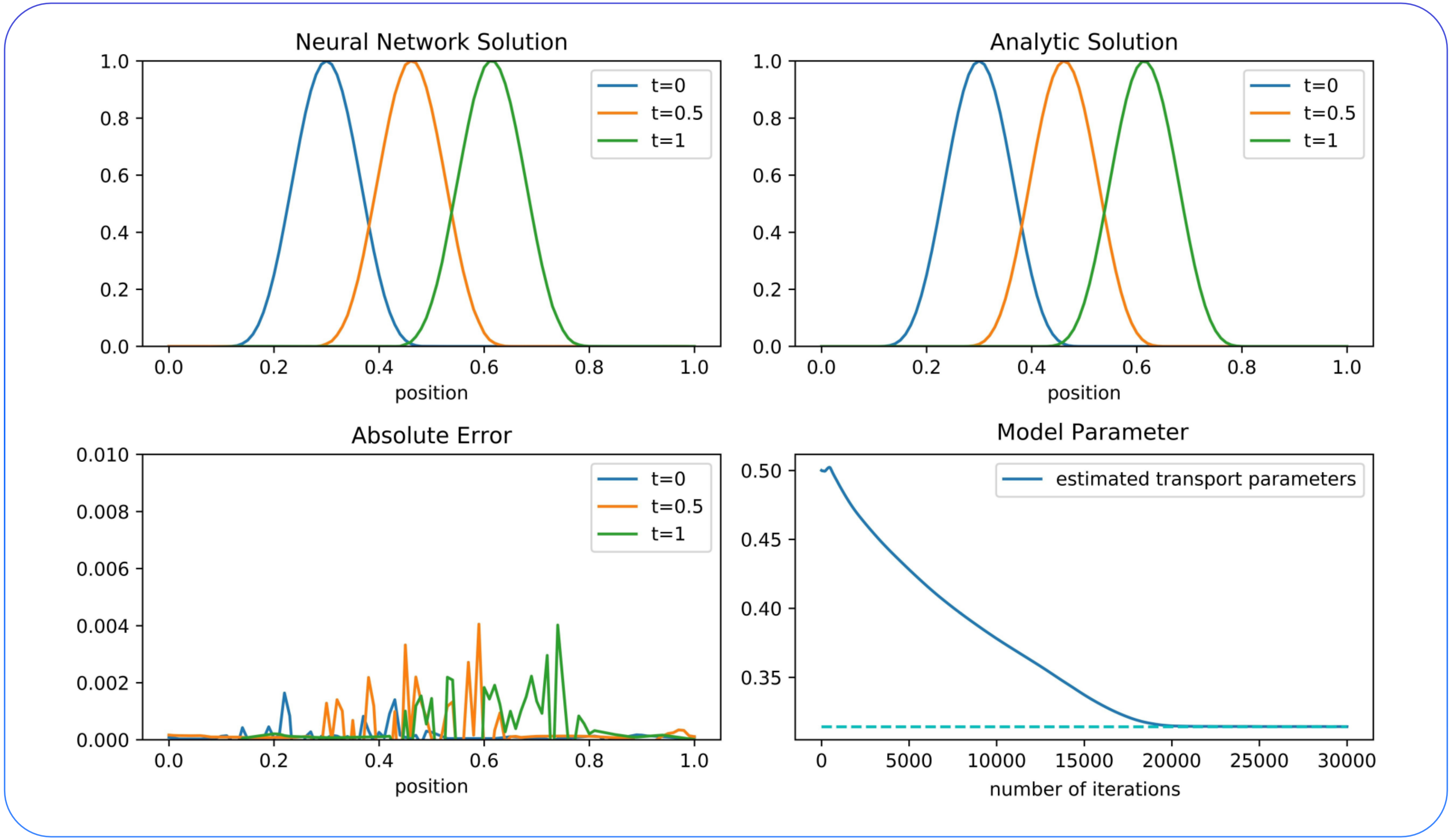


- Detection of Changes in the Number of Confirmed Cases Approximately 4 Days in Advance
- Analysis of the Impact of Prevention Policies on Containing the Spread

Experiments: Transport equation

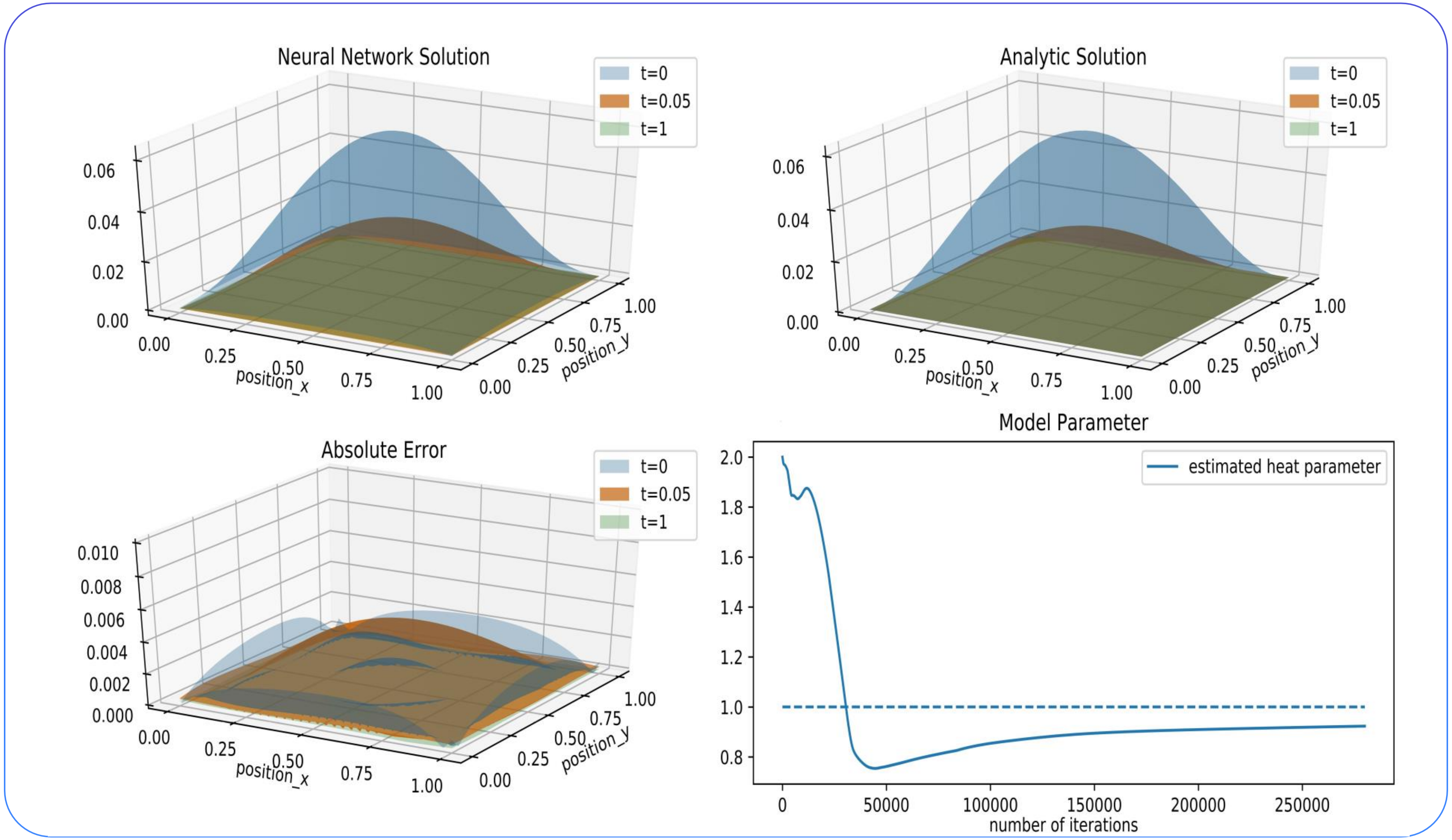
- Experimental result for 1D transport equation





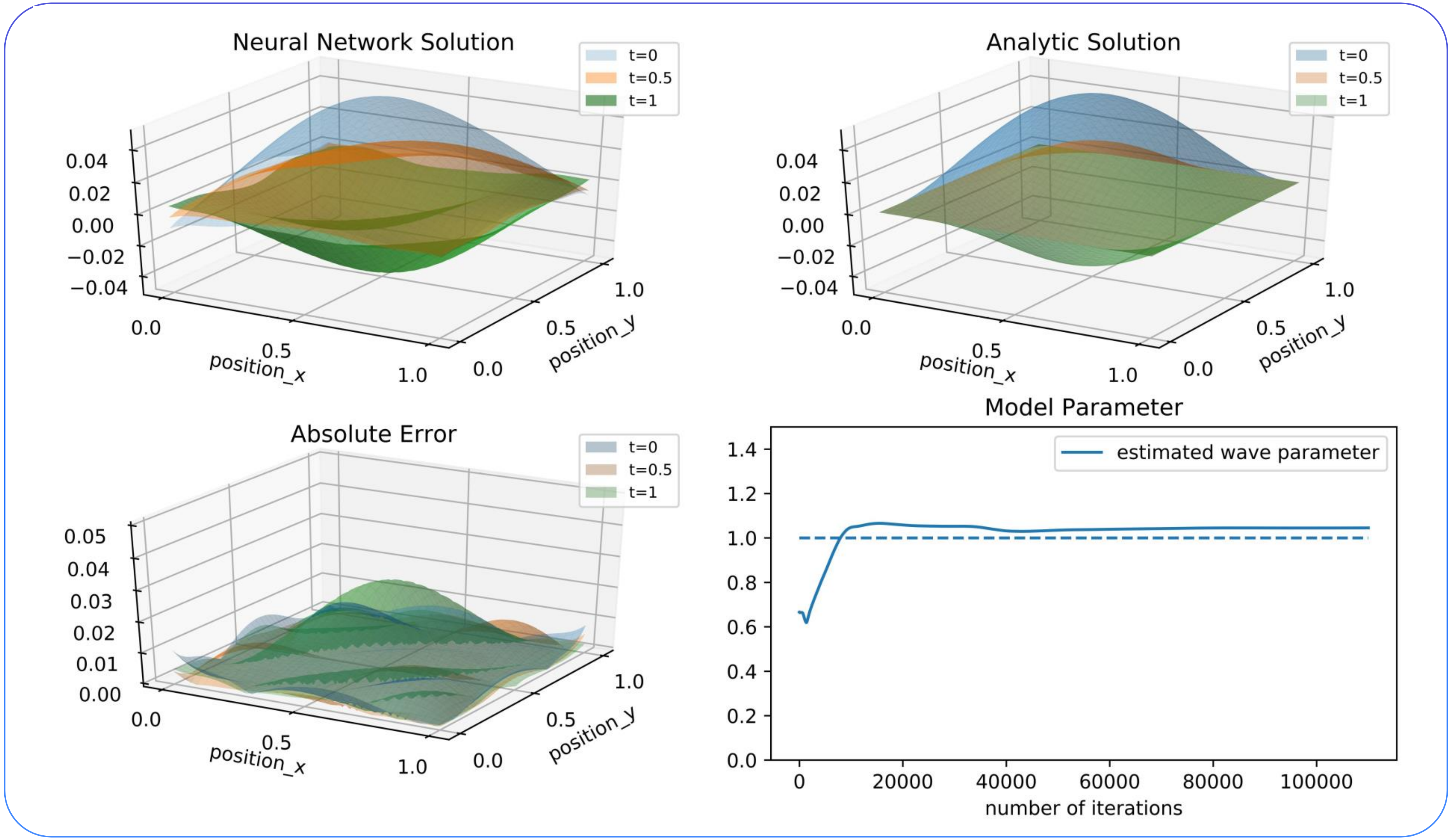
Experiments: Heat equation

- Experimental result for 2D heat equation with $u(0,x,y)=x(1-x)y(1-y)$



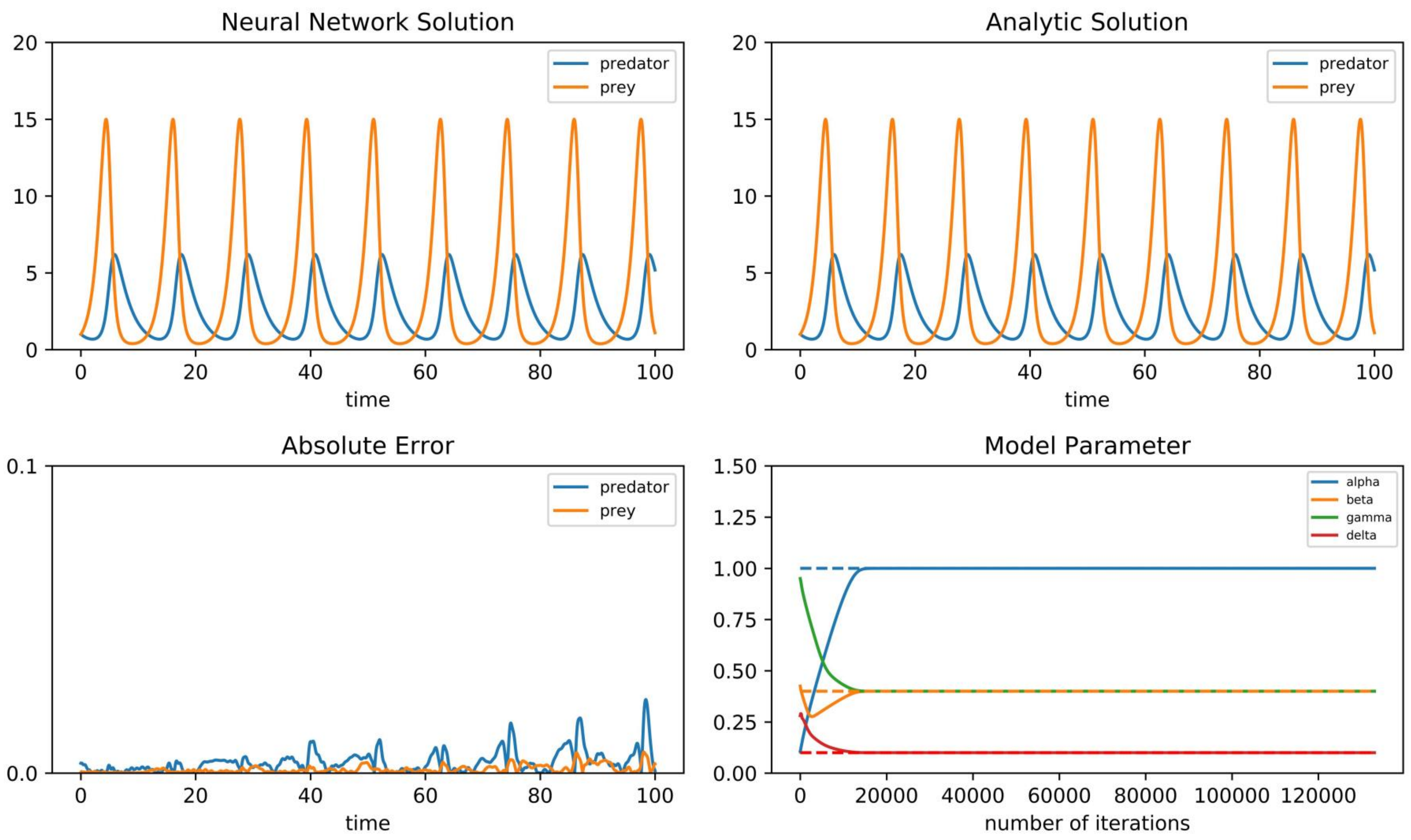
Experiments: Wave equation

- Experimental result for 2D wave equation

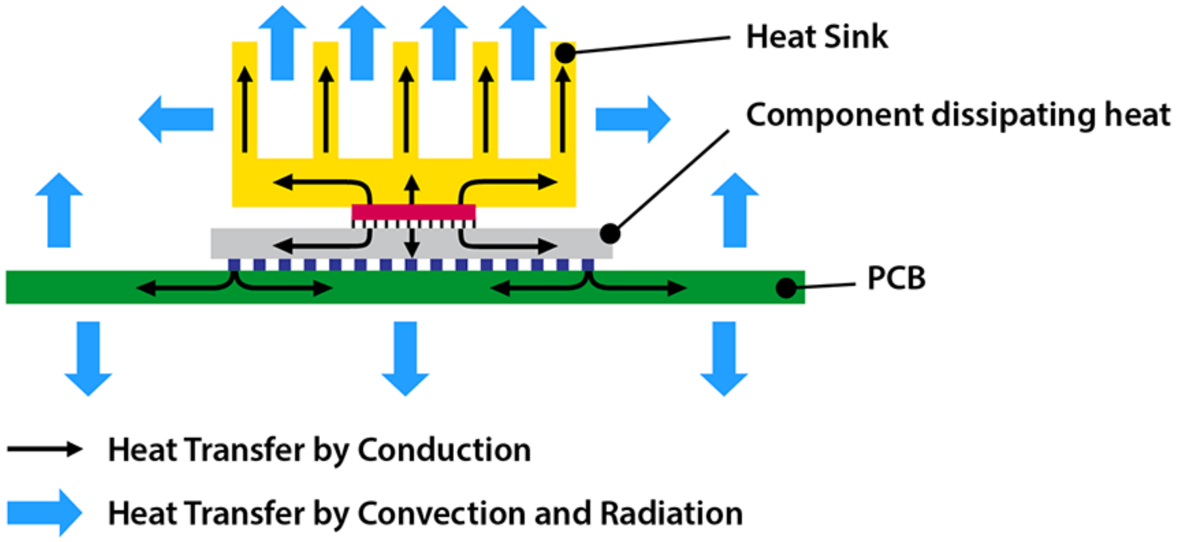


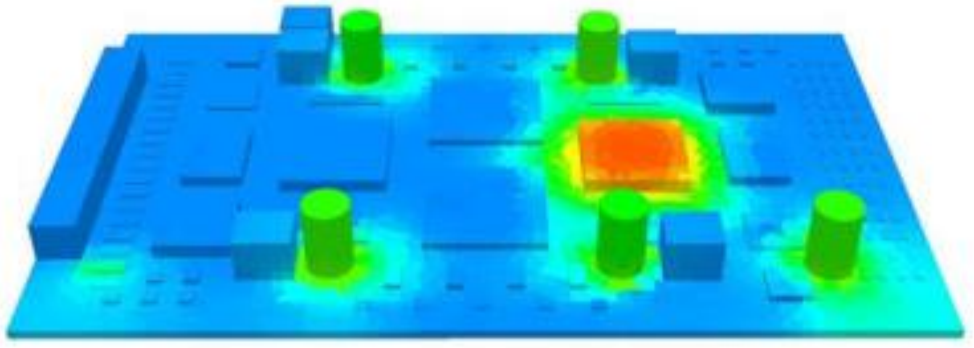
Experiments: Lotka-Volterra

- Experimental result for Lotka-Volterra equation



Semiconductor Heat Management

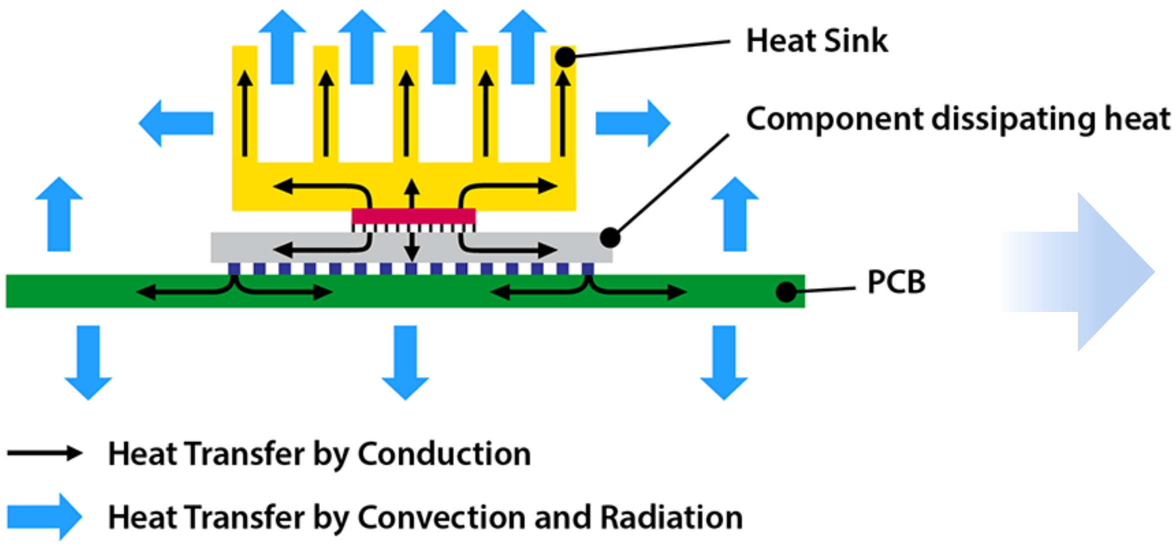


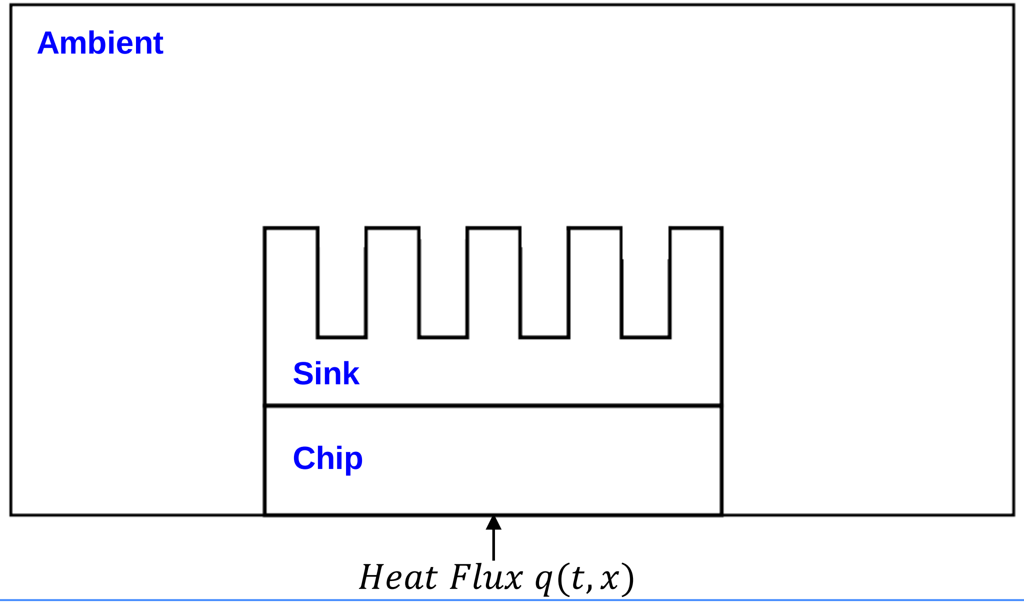


- Semiconductor chips (ICs) are the main source of heat in electronic equipment
- Overheating is a major cause of chip failure, directly affecting the lifespan of the chip
- As semiconductor chip sizes continue to shrink, thermal management becomes increasingly critical
- Proper chip placement and cooling strategies are essential to prevent critical components from overheating and failing

Physics → Model

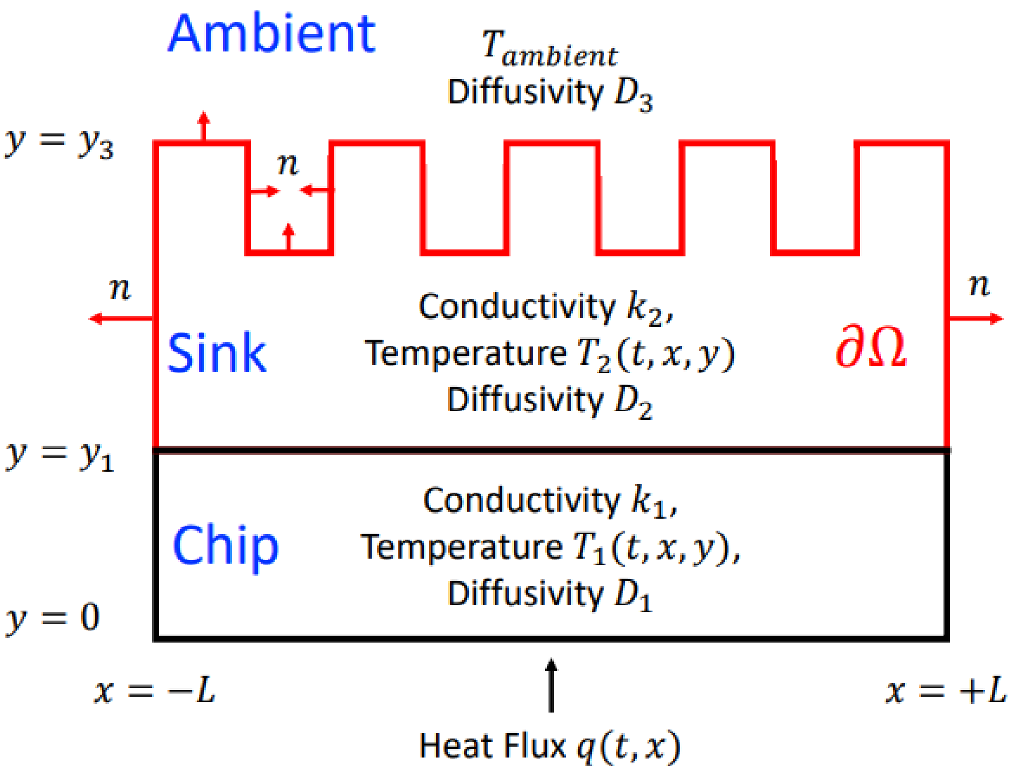
Semiconductor Heat Management - Heat Sink Based Heat Spreading Model





- Semiconductor heat dissipation model through a heat sink
- Verifying the feasibility of simulation using PINN through simple modeling

Semiconductor Heat Management - Heat Sink Based Heat Spreading Model



$\frac{d}{dt}T_i = D_i \Delta T_i$, for $i = 1, 2, 3$

$-k_1 \frac{\partial}{\partial y} T_1 = q(t, x)$, at $y = 0$

$-k_1 \frac{\partial}{\partial y} T_1 = -k_2 \frac{\partial}{\partial y} T_2 = h_1(T_1 - T_2)$, at $y = y_1$

$-k_2 \frac{\partial}{\partial n} T_2 = h_2(T_2 - T_{ambient})$, at $\partial\Omega$

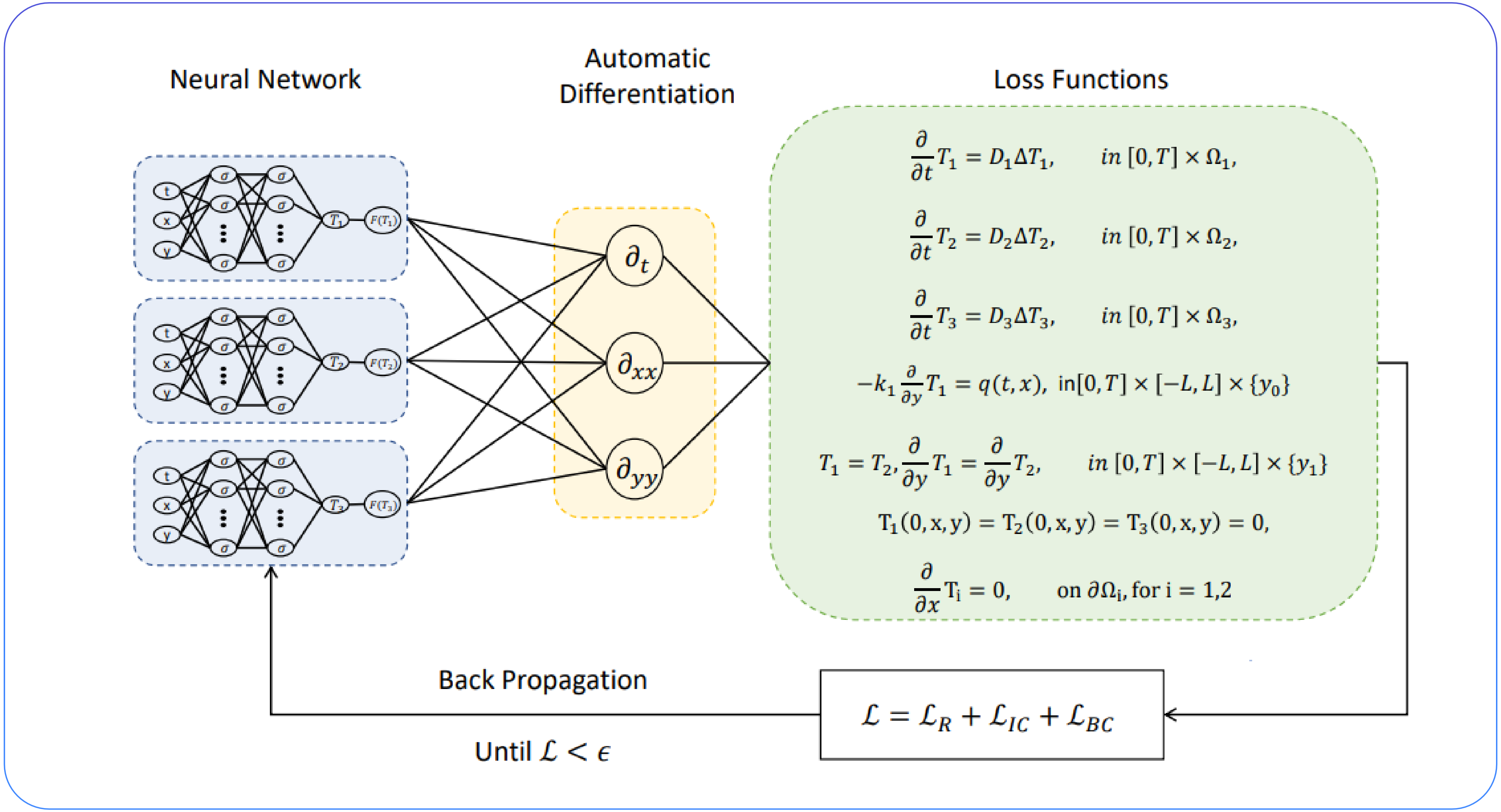
$\frac{\partial}{\partial_x} T_1 = 0$, at $x = \pm L$, $y_0 \le y \le y_1$

B.C

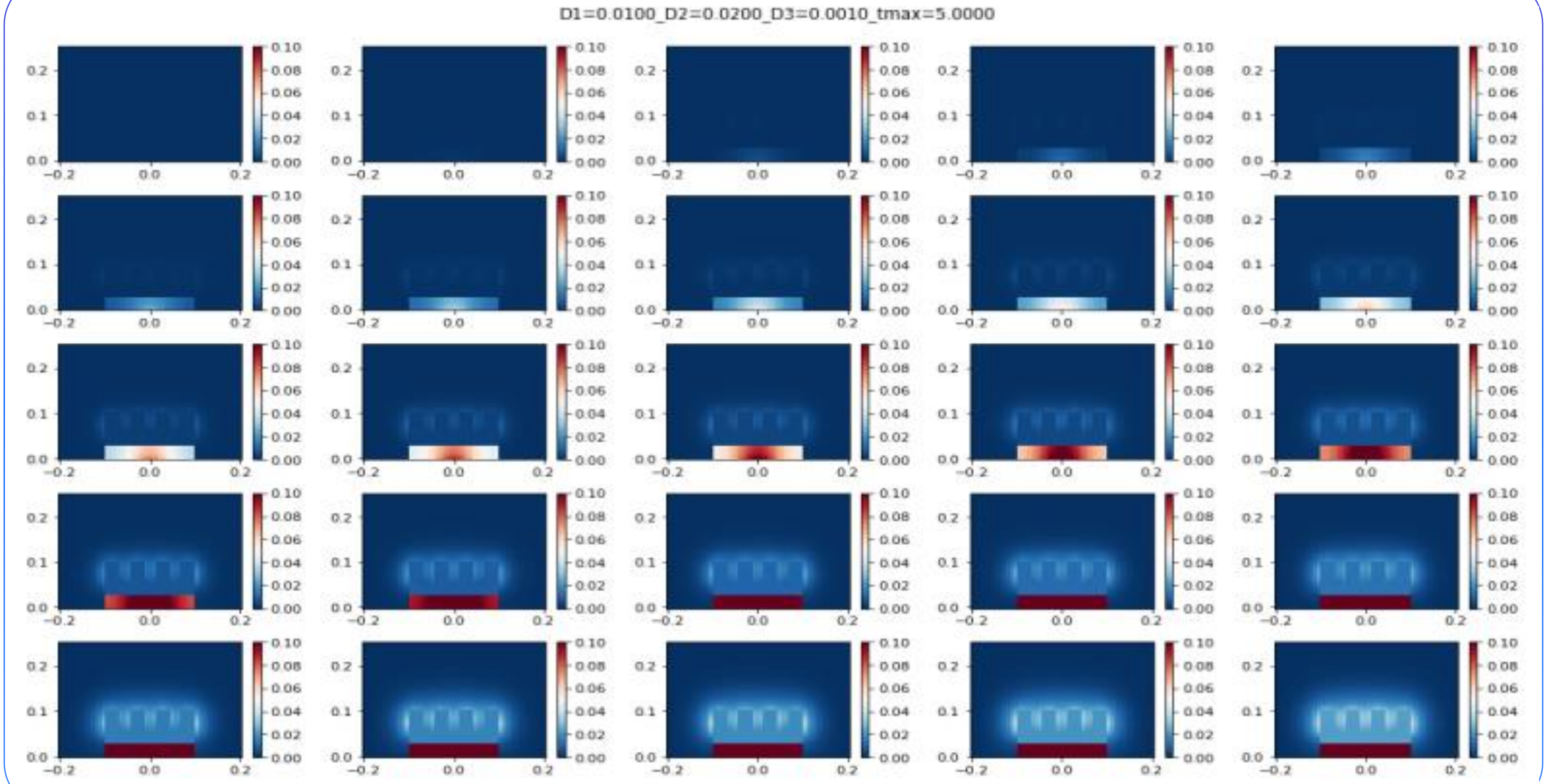
$T_1 = T_2 = T_{ambient} = 0$, at $t = 0$

- Modeling using the problem of solving PDEs for three systems: Chip, Sink, and Ambient

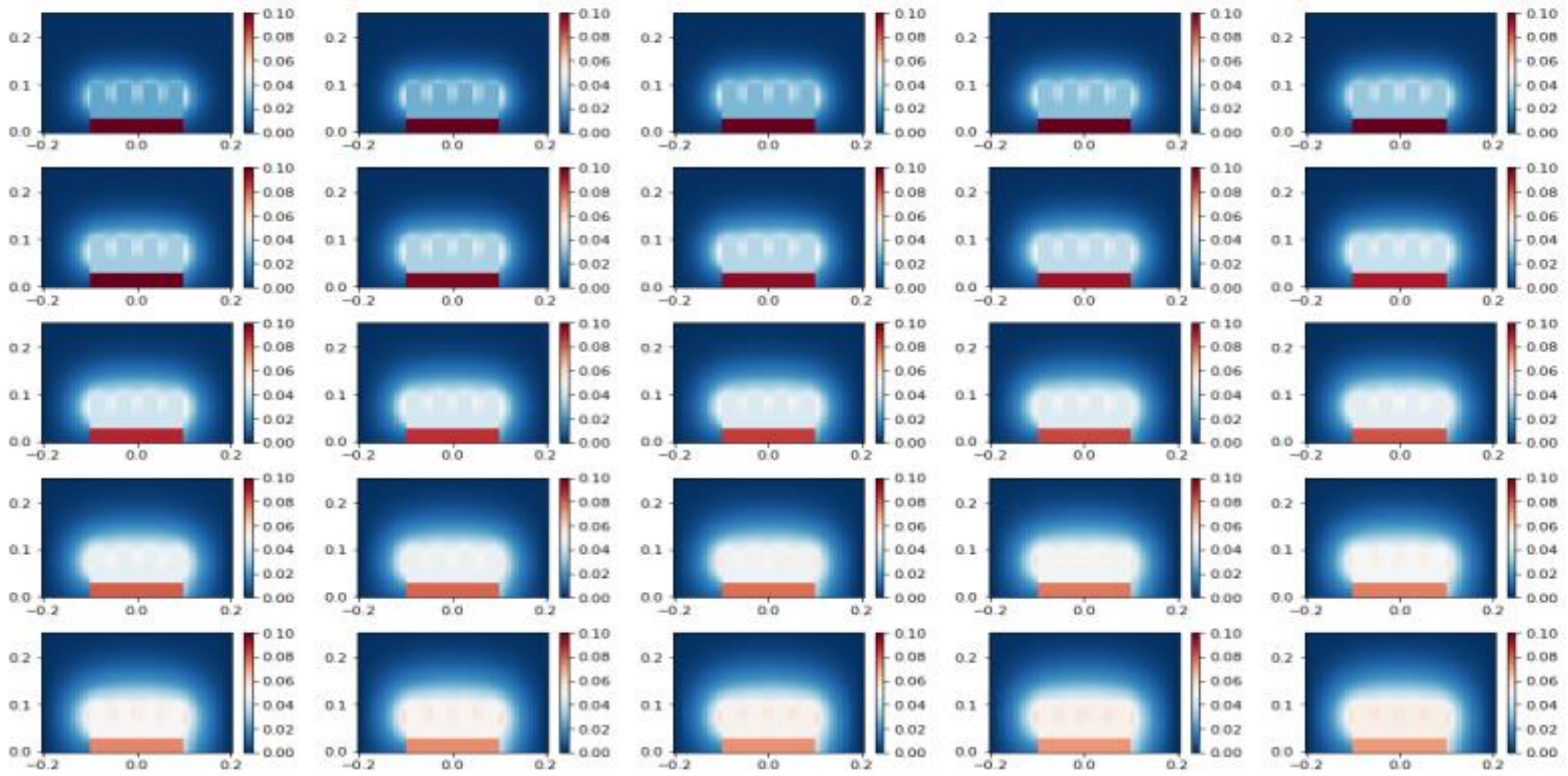
Semiconductor Heat Management – PINN Structure



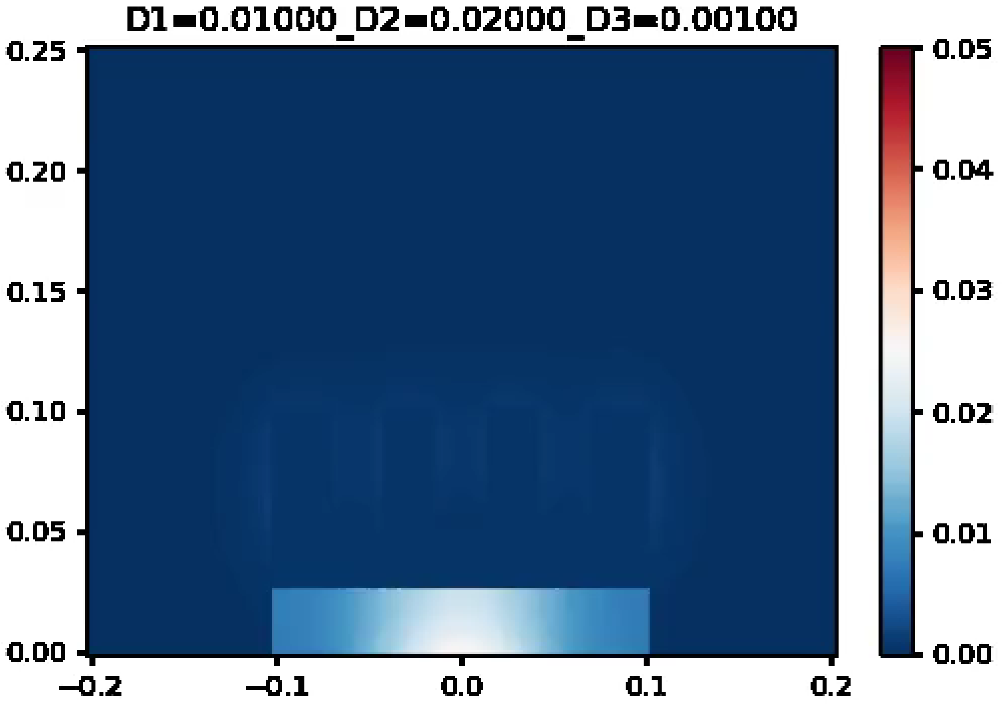
Semiconductor Heat Management – PINN Simulation Results



Semiconductor Heat Management – PINN Simulation Results(continued)

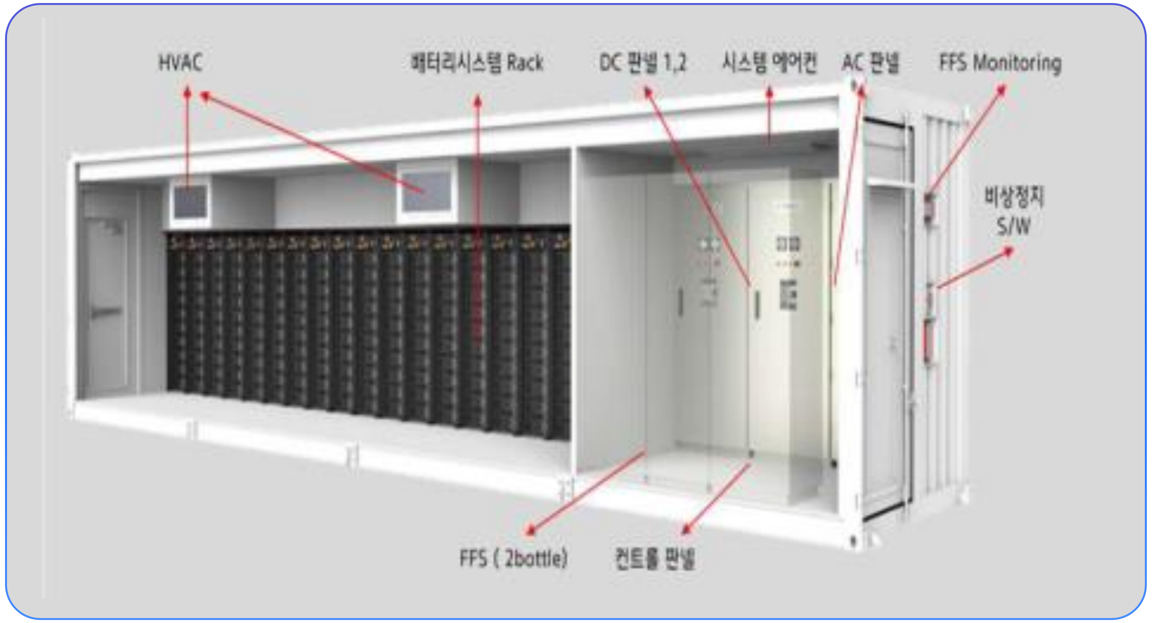


Semiconductor Heat Management – PINN Simulation Results



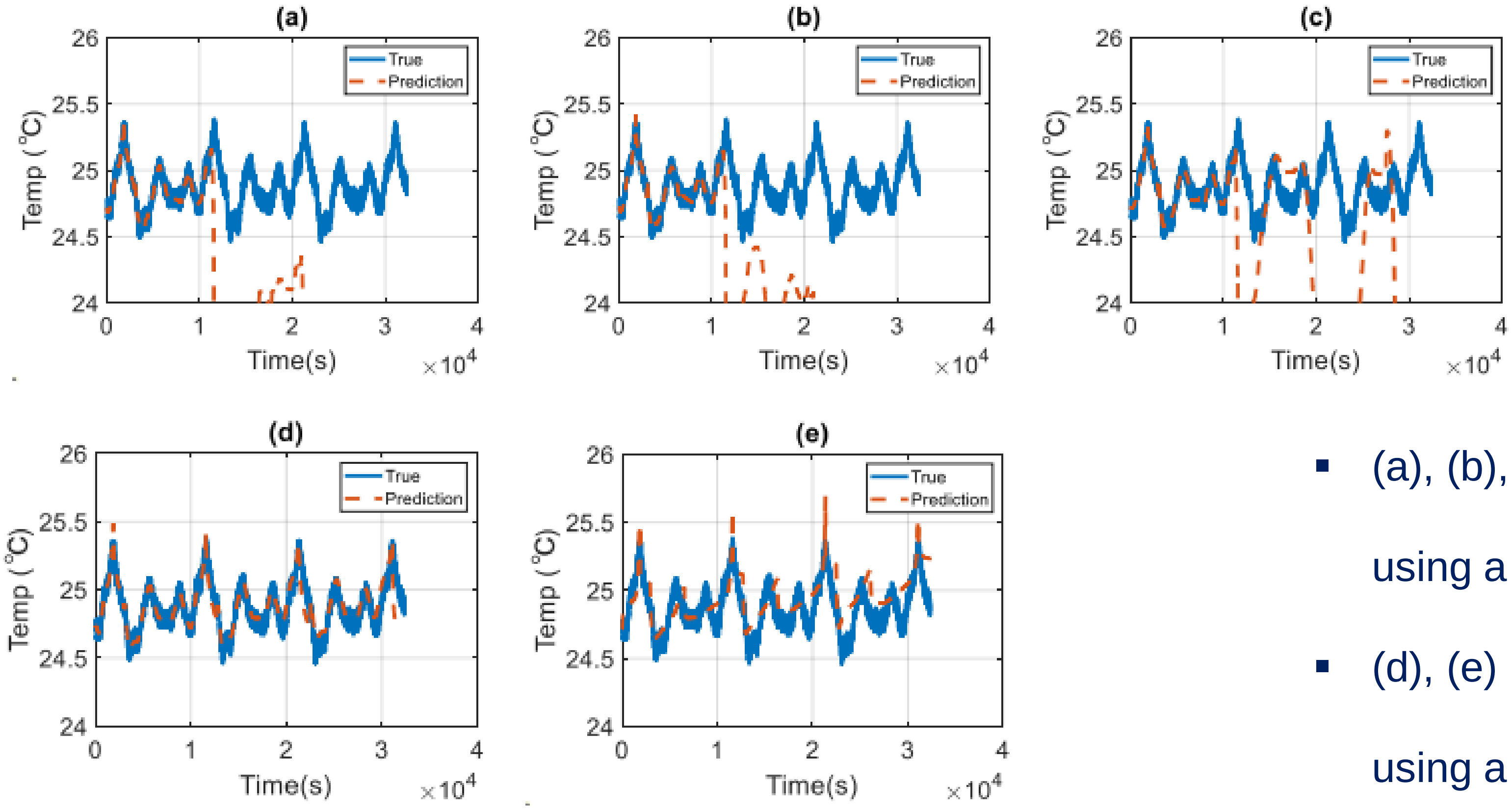
Applications of Neural Simulators - Predicting Battery Temperature and Lifespan

- Lithium-ion batteries emit heat during repeated charging and discharging cycles.
- This process causes degradation, affecting the battery's lifespan.
- Accurately predicting battery lifespan allows for precise timing of battery replacement.
- Managing the lifespan of individual batteries is especially challenging in large battery cell containers, not just simple single batteries



Applications of Neural Simulators - Predicting Battery Temperature and Lifespan

Battery Temperature Prediction Results Solved Using the PINN Inverse Problem Concept

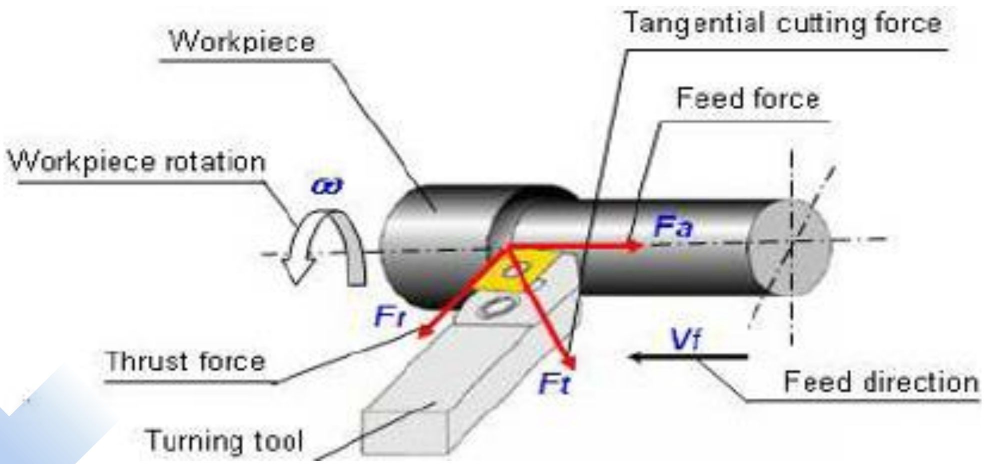


- (a), (b), (c) perform temperature prediction using a basic ANN structure.
- (d), (e) perform temperature prediction using a PINN structure.

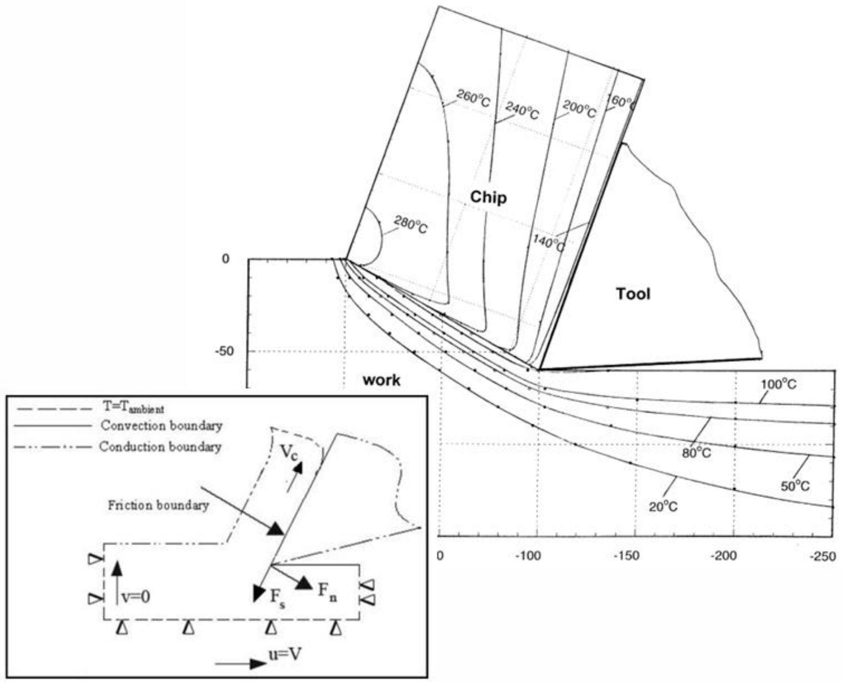
Cho, G. (2022). *Artificial Neural Network Methods for Lithium-Ion Battery Behavior Predictions* (Doctoral dissertation).

Metal Cutting

Mechanical System



Mathematical Model



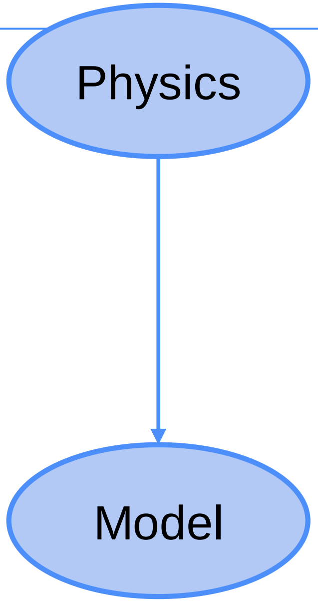
$$Q_S = W_c = F_V \cdot V$$

$$Q_r = \frac{F_{fr} \cdot V}{\lambda_h}$$

Heat Generation

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

Governing Equation



Metal Cutting

Mathematical Model

$$Q_S = W_c = F_V \cdot V$$

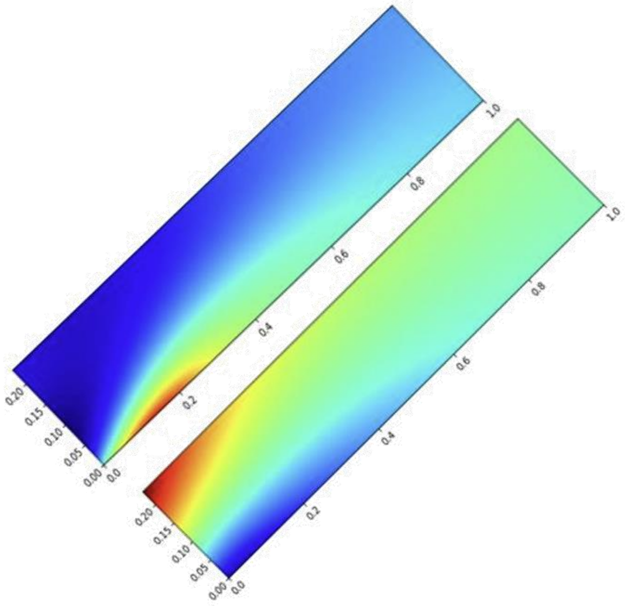
$$Q_r = \frac{F_{fr} \cdot V}{\lambda_h}$$

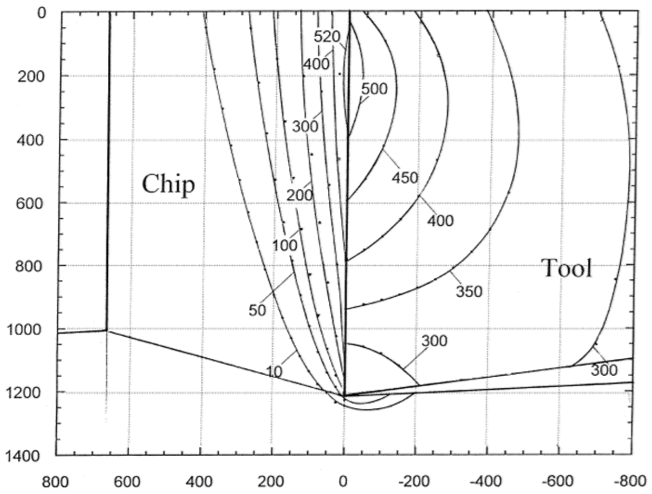
Heat Generation

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

Governing Equation

PINN Result





- It allows for rapid prediction of temperature distribution as the material, process conditions, and thermal conditions change

Part II. Neural PDE Solvers

Operator Learning

Galerkin Transformer

Neural Network

Matrix Multiplication

$$y_i = \sum_j K(i,j)u_j$$

Neural Operator

Kernel Integration

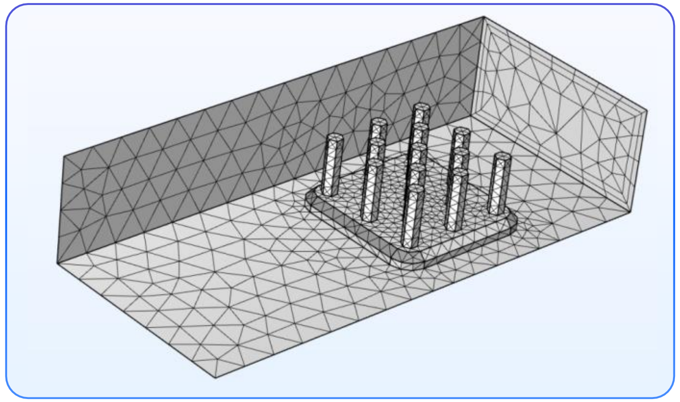
$$y(x) = \int K(x,y)u(y)dy$$

Advantages

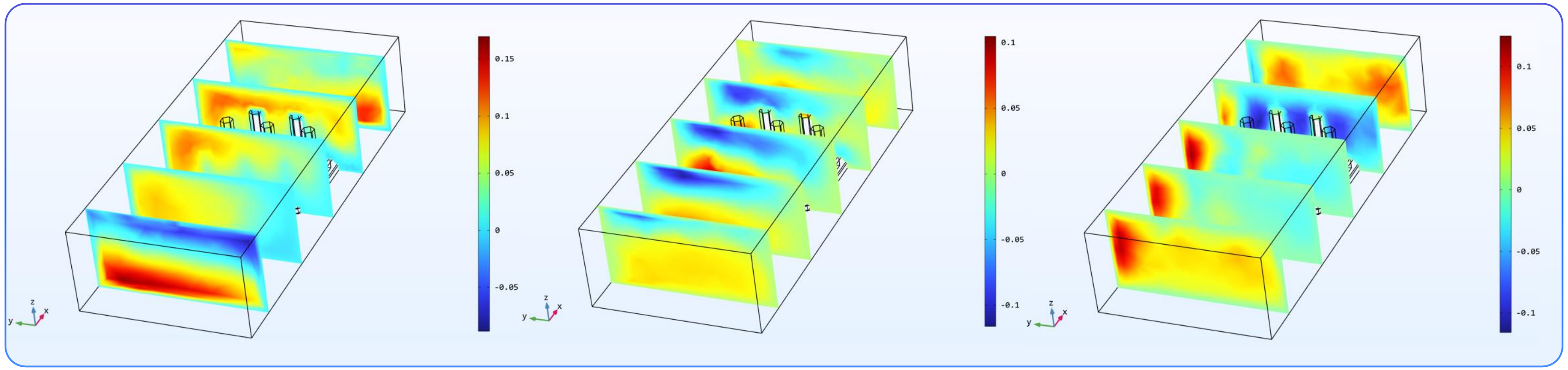
- Can be applied to irregular meshes
- Broad range of applications

Disadvantages

- Vulnerable to overfitting
- Few comparative experimental results



Heatsink simulation



Fourier Neural Operator

Neural Network

Convolution

$$y_i = \sum_j K(i - j) u_j$$

Neural Operator

Kernel Integration

$$y(x) = \int K(x - y) u(y) dy$$

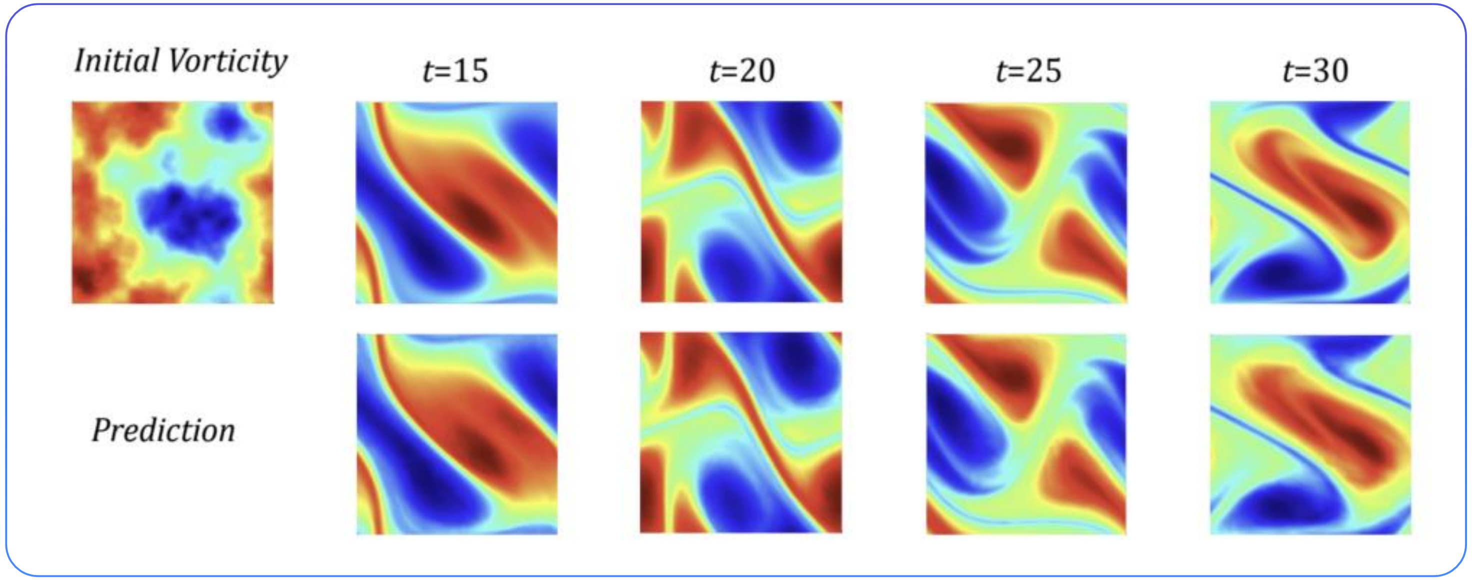
Advantages

- High performance on regular mesh data
- Many comparative experimental results

Disadvantages

- Higher computational complexity than Galerkin Transformer
- Difficult to apply to irregular meshes

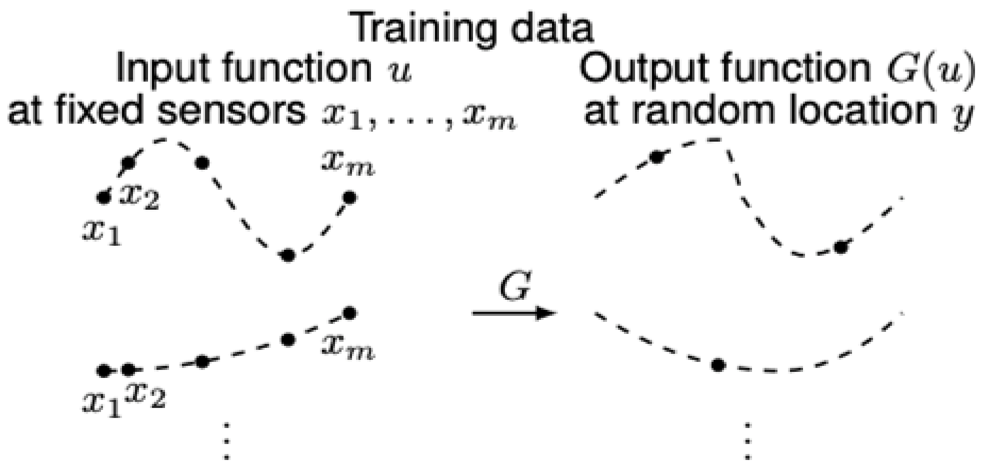
Navier-Stokes Simulation



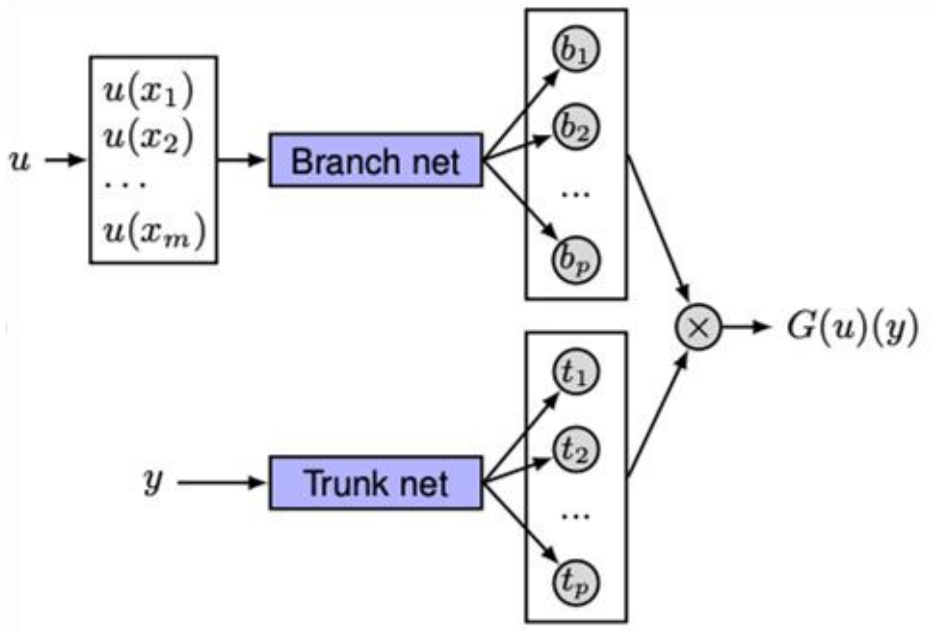
Deep Operator Network(DeepONet)

Learning an operator $G: u(x) \rightarrow G(u)(y)$ using the network

Data

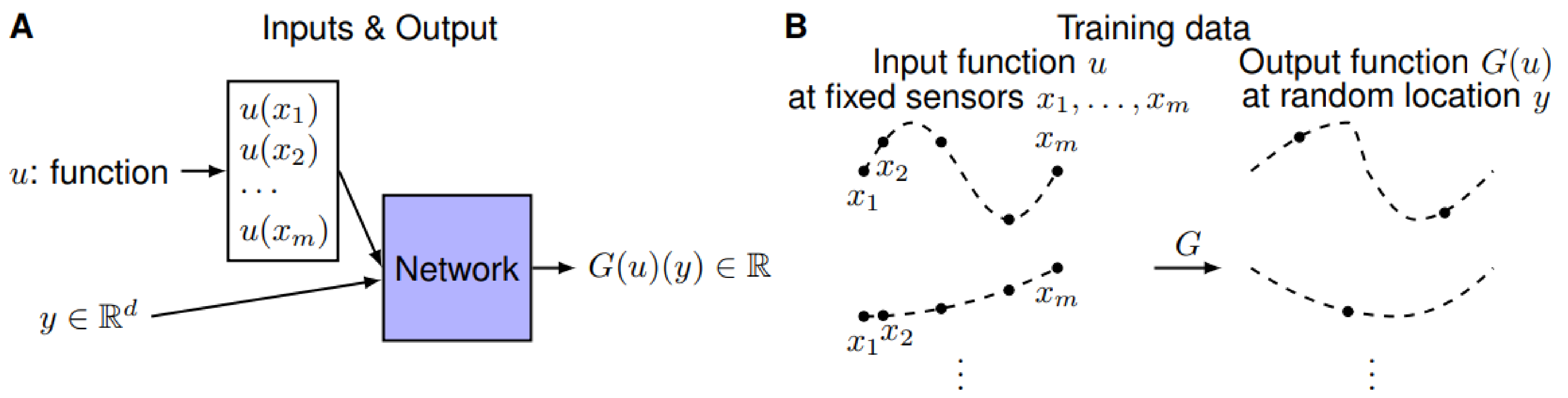


Model



Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature machine intelligence*, *3*(3), 218-229.

Deep Operator Network(DeepONet)



- A network trained to infer the value at a desired location y based on the measurements$\{u(x)\}$

Advantages

- Easy to implement
- First proposed model
- Abundant theoretical analysis resources

Disadvantages

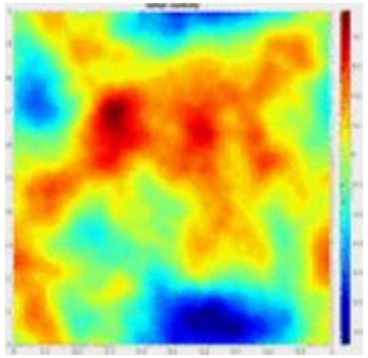
- Lower accuracy compared to other recent models (FNK, gk-Transformer)

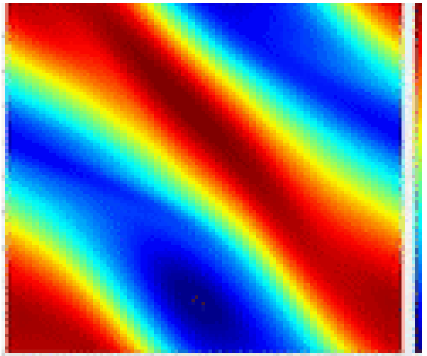
Applications of Neural Simulators – Navier-Stokes Simulation

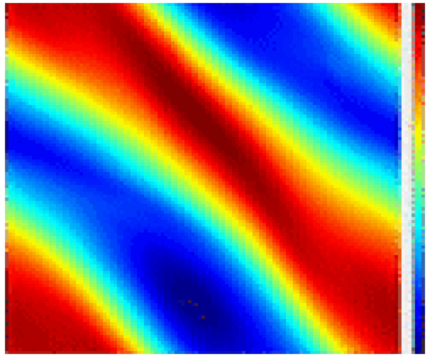
- 100 Times Faster Analysis Possible

Initial Condition | Ground Truth | Prediction

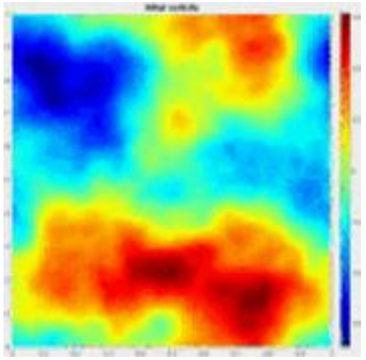
Case 1

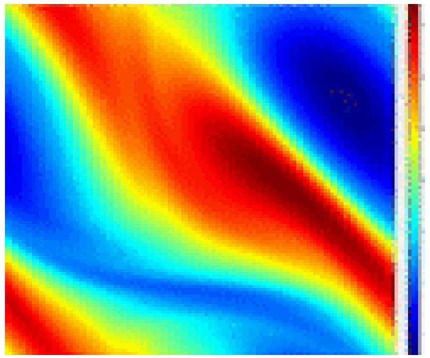


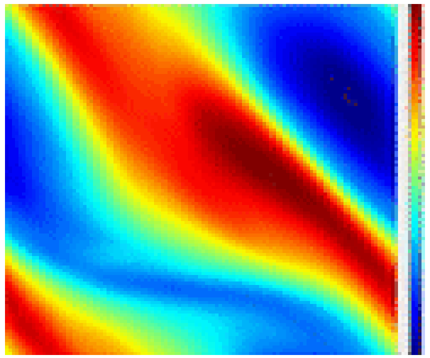




Case 2







MATLAB EXPO

Thank you for your attention

Hyung Ju Hwang, POSTECH

hjhwang@postech.ac.kr

MathWorks®