Visual analysis to support the derivation and solution of PDEs using neural networks

Jan 20, 2022

Academic Center for Computing and Media Studies
Kyoto University
Koji KOYAMADA

Beijing Olympics

- Two weeks until the Beijing Olympics.
- Preparations are in full swing under strict infection control measures.

Previous ChinaVis conferences

The 9th China Visualization and Visual Analytics Conference, Xining, July 24-27, 2022

The 8th China Visualization and Visual Analytics Conference, Wuhan, July 24-27, 2021
The 7th China Visualization and Visual Analytics Conference, Xi'an, July 18-21, 2020
The 6th China Visualization and Visual Analytics Conference, Chengdu, July 21-24, 2019
The 5th China Visualization and Visual Analytics Conference, Shanghai, July 26-28, 2018
The 4th China Visualization and Visual Analytics Conference, Qingdao, July 17-19, 2017
The 3rd China Visualization and Visual Analysis Conference, Changsha, July 21-23, 2016
The 2nd China Visualization and Visual Analytics Conference, Tianjin, July 17-18, 2015
The 1st China Visualization and Visual Analytics Conference, Beijing, July 19-20, 2014
4th Visualization Workshop 2013, Beijing, July 12-13, 2013
2011 3rd Visualization Symposium, Beijing, July 23, 2011
2009 Second Visualization Symposium, Beijing, April 23, 2009

2008 First Visualization Workshop, Beijing, June 24, 2008

Visualization plays a more important role in achieving “together for a shared future"

Chinavis

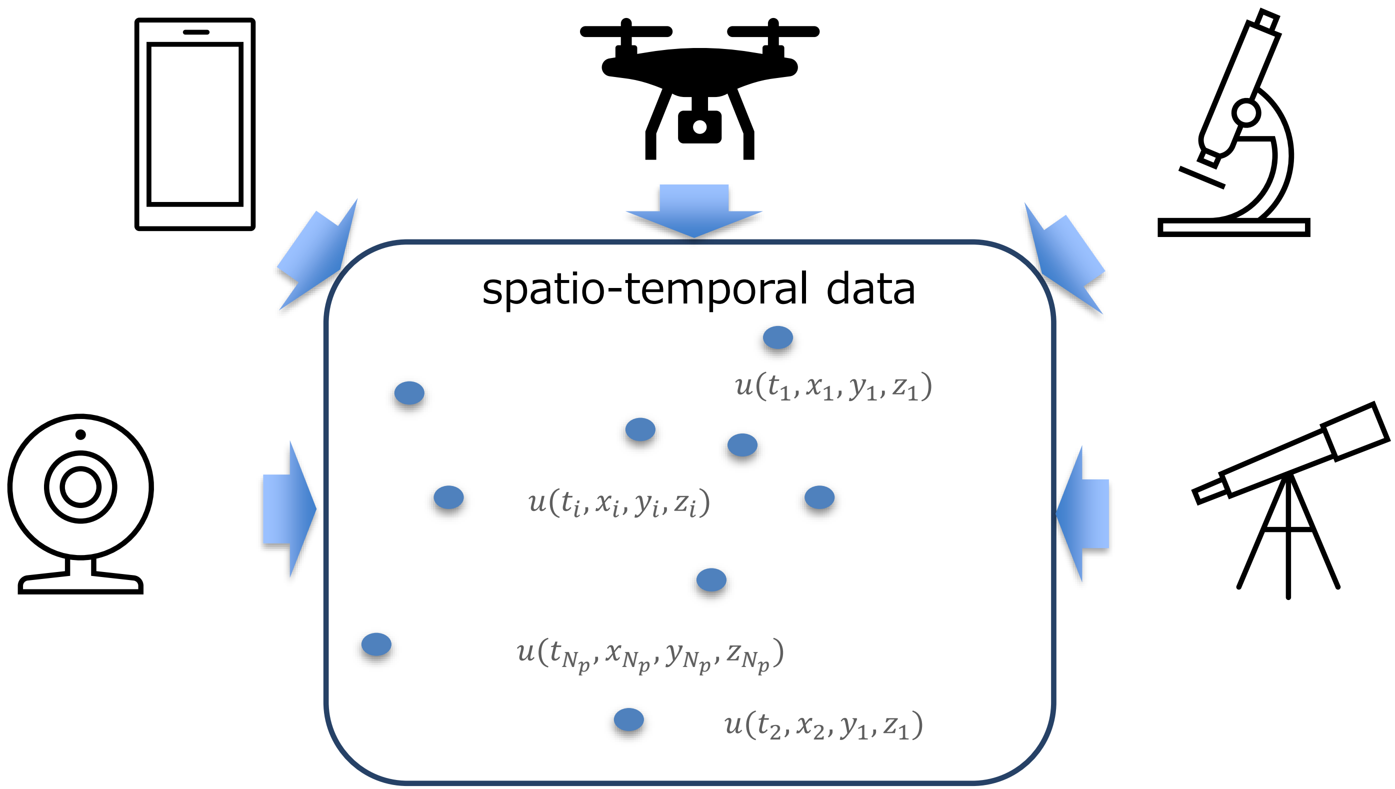
- It aims to improve the communication of the Visualization and Visual Analytics communities in China and surrounding regions.
- It is a strong international conference (acceptance rate of 29.4%(42/143, 2021))

Content

How does our visual data science research meet AI?

- AI-enhanced visualization (AI4VIS)
  1. 3-D book data page segmentation and extraction
  2. Visualization of plasma shape in the lhd-type helical fusion reactor
- Surrogate model
- Visualization-enhanced AI (VIS4AI)
  1. PDE derivation from spatio-temporal data
  2. PDE solution from spatio-temporal data

Point data in a spatio-temporal space



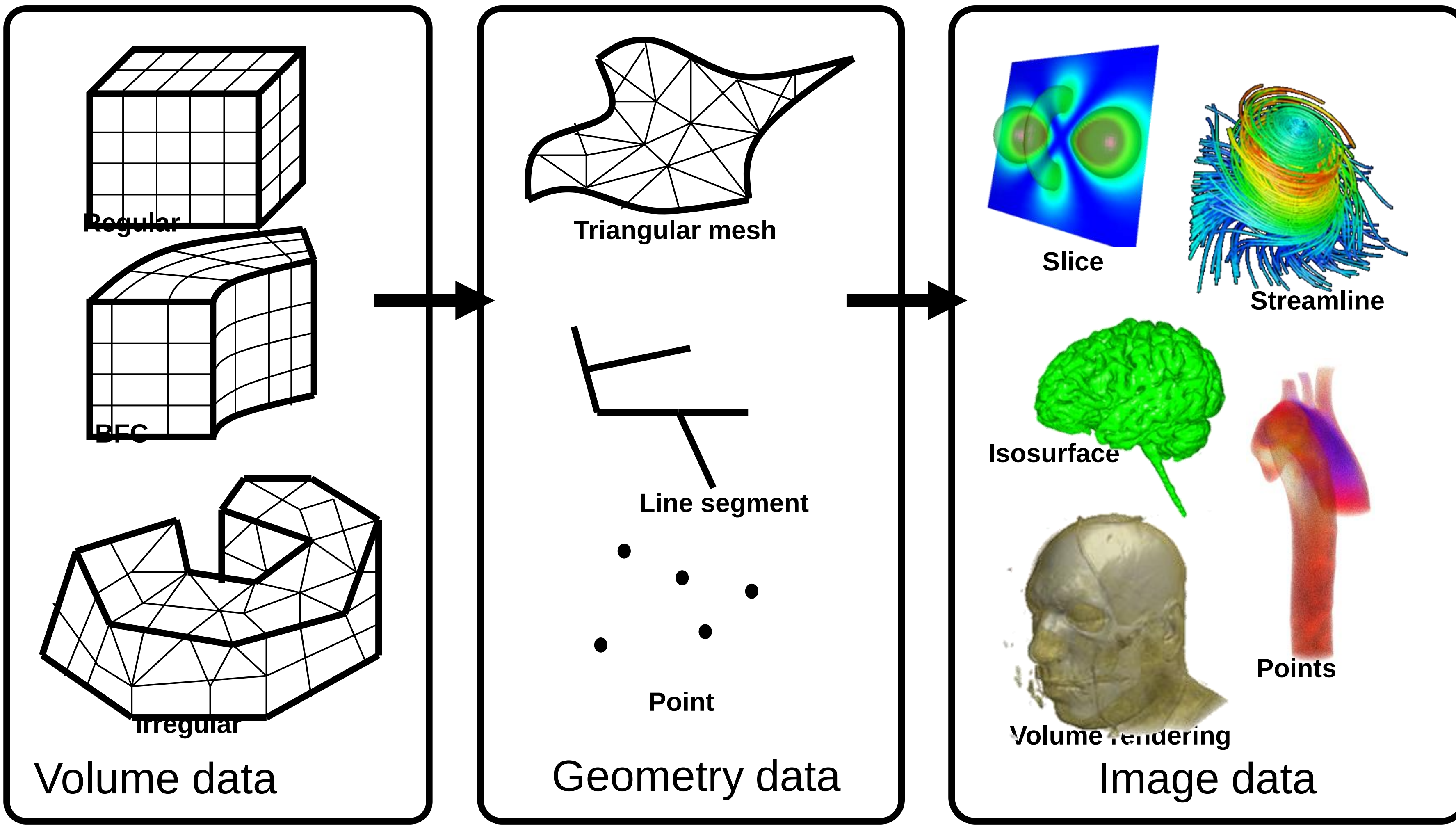
- Physical quantity measured at a certain spatial position at a certain fixed time

$$u\left(t_i, x_j, y_j, z_j\right), i = 1, N, j = 1, M$$

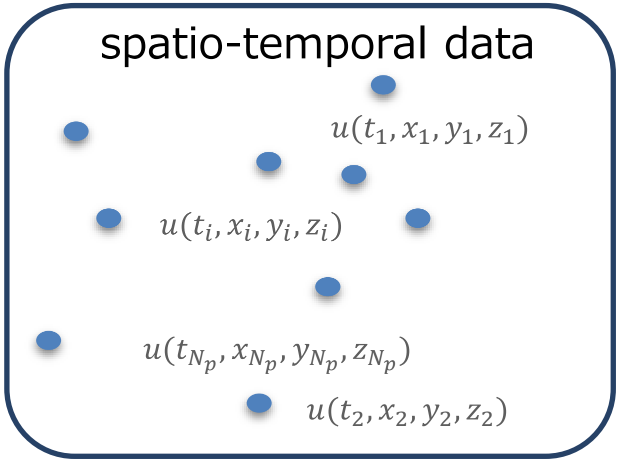
- Data obtained at a certain spatiotemporal position

$$u(t_i, x_i, y_i, z_i), i = 1, N$$

Data visualization

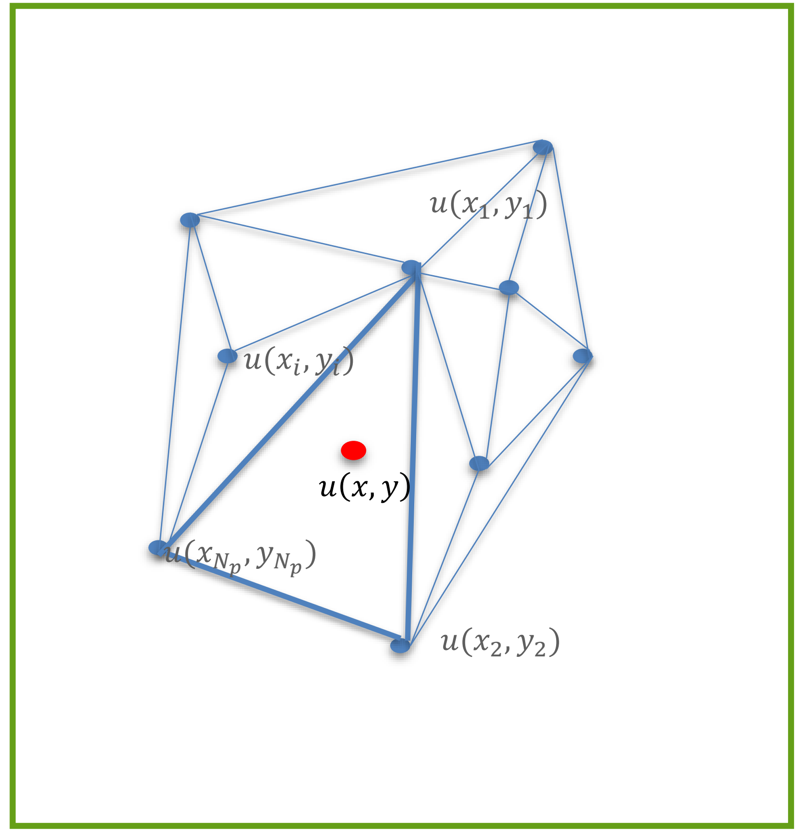


Local interpolation



Generation of mesh geometry

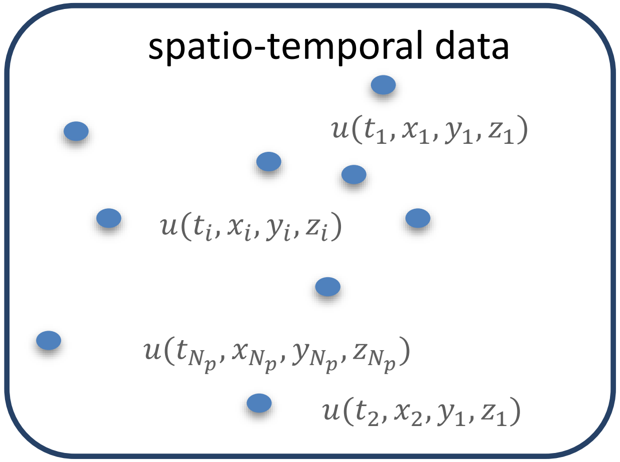
Location of a given point

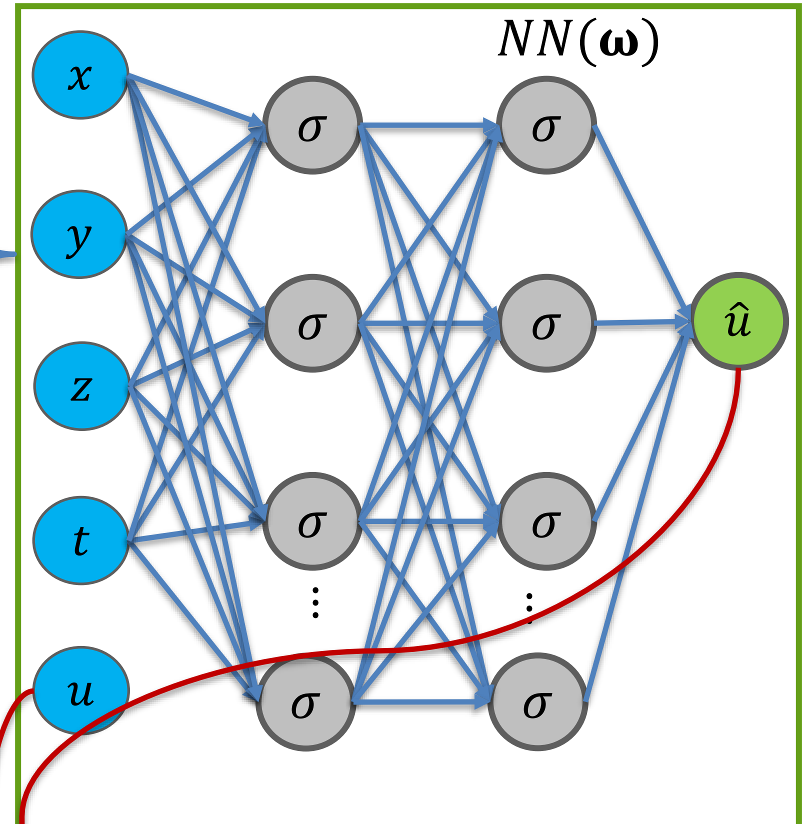


If the point, $u(t,x,y,z)$, is located inside the j-th simplex, then

$$u(t,x,y,z) = \sum_{i=1}^{5} N_j(t_i, x_i, y_i, z_i) u_j(t_i, x_i, y_i, z_i)$$

Global approximation





Regression of spatio-temporal data as input

Parameter approximation minimizing a loss function

Neural network(NN) model

$$argmin\{MSE_u(\boldsymbol{\omega})\}$$

$$MSE_u = \frac{1}{N_u}||u - \hat{u}||_2^2$$

$$u(t,x,y,z) = u_j^{L+1} = \sigma^L\left(\sum_{i=1}^{n_L} w_{i,j}^L \sigma^{L-1}\left(\sum_{i=1}^{n_{L-1}} w_{i,j}^{L-1} \cdots \sigma^0\left(\sum_{i=1}^{n_0} w_{i,j}^0 u_i^0\right) \cdots\right)\right)$$

Universal Approximation Theorem in Neural Networks

G.Cybenko ,"Approximation by Superpositions of a Sigmoidal Function," 1989

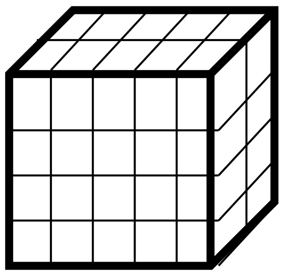
Abstract. In this paper we demonstrate that finite linear combinations of compositions of a fixed, univariate function and a set of affine functionals can uniformly approximate any continuous function of *n* real variables with support in the unit hypercube; only mild conditions are imposed on the univariate function. Our results settle an open question about representability in the class of single hidden layer neural networks. In particular, we show that arbitrary decision regions can be arbitrarily well approximated by continuous feedforward neural networks with only a single internal, hidden layer and any continuous sigmoidal nonlinearity. The paper discusses approximation properties of other possible types of nonlinearities that might be implemented by artificial neural networks.

Key words. Neural networks, Approximation, Completeness.

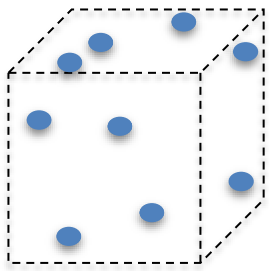
NN model visualization

To visualization of NN models in a spatio-temporal space,

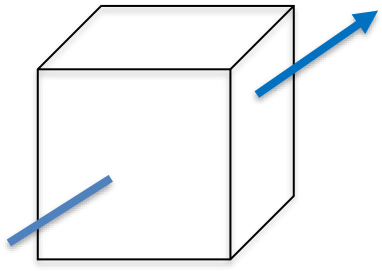
1. Regular grid generation
   - Prepare grids with an adequate resolution
   - Evaluate a function value at each grid point
   - Employ Marching cubes



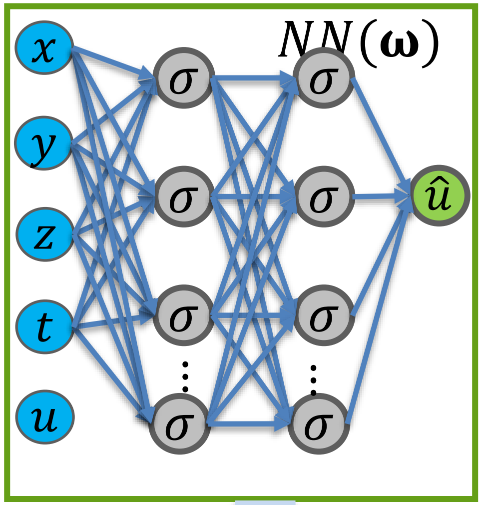
2. Render the NN model using particles
   - Sample points in a spatio-temporal space
   - Employ particle-based volume rendering



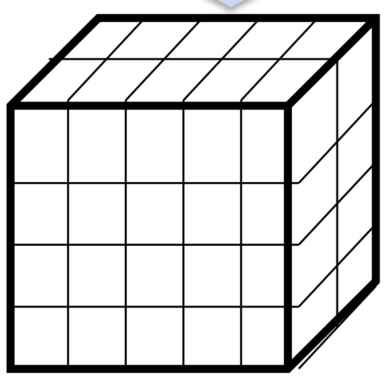
3. Ray-trace the NN model without grids
   - Cast a ray in a spatio-temporal space
   - Integrate the NN model along the ray

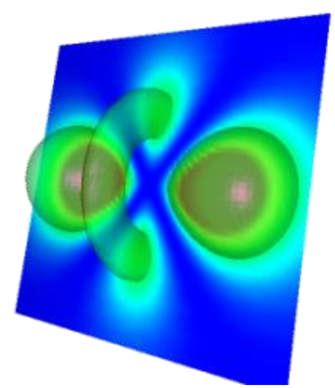


Regular grid generation




```
volume = []
for i in range(0, 10):
 for j in range(0, 10):
  for k in range(0, 10):
   for l in range(0, 10):
     volume.append(u(i, j, k, l))
```

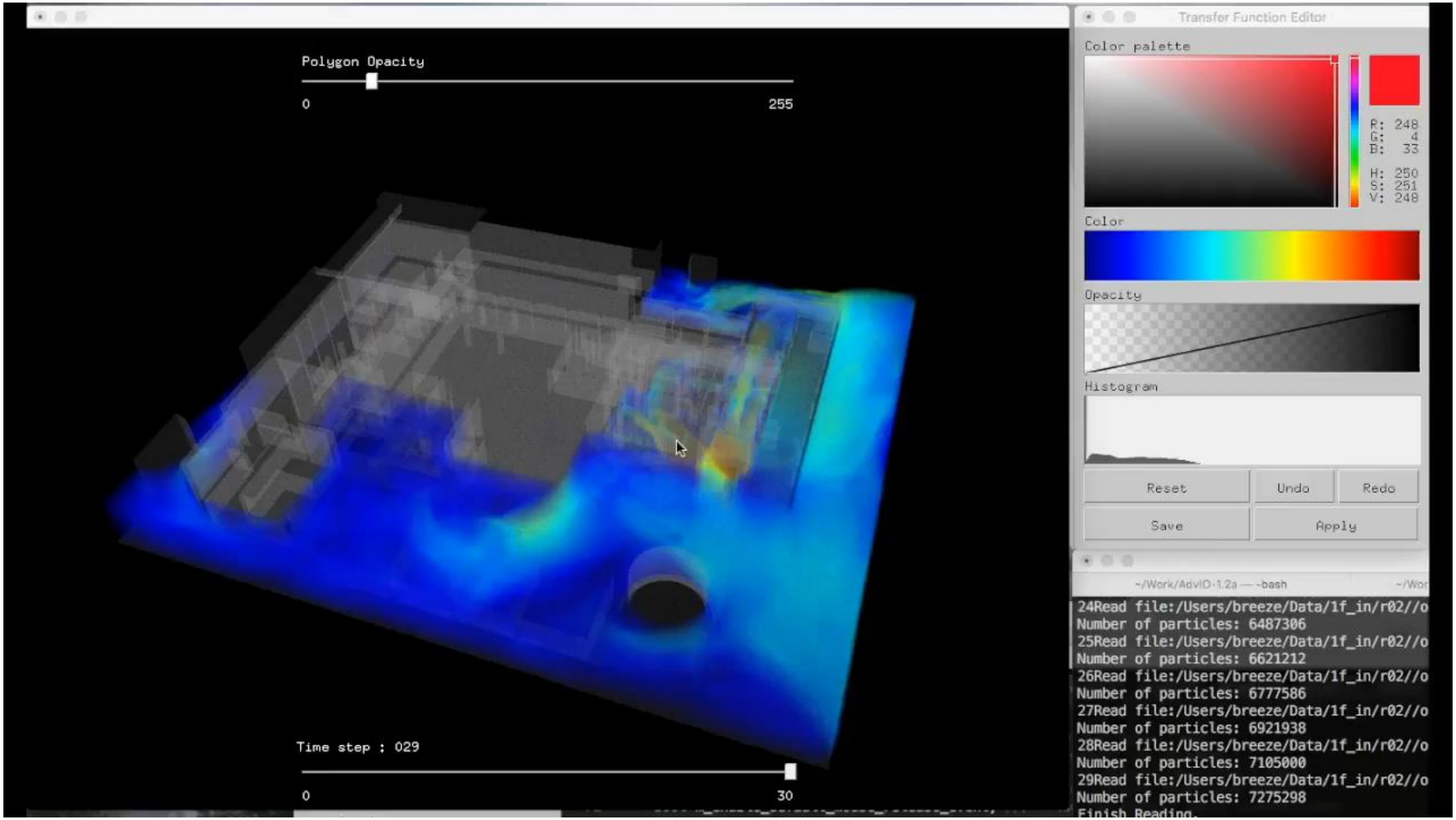




$$\boldsymbol{u(t,x,y,z) = \sigma^{L}\left(\sum_{i=1}^{n_L} w_{i,j}^{L}\sigma^{L-1}\left(\sum_{i=1}^{n_{L-1}} w_{i,j}^{L-1}\cdots\ \sigma^{0}\left(\sum_{i=1}^{n_0} w_{i,j}^{0}u_i^{0}\right)\cdots\right)\right)}$$

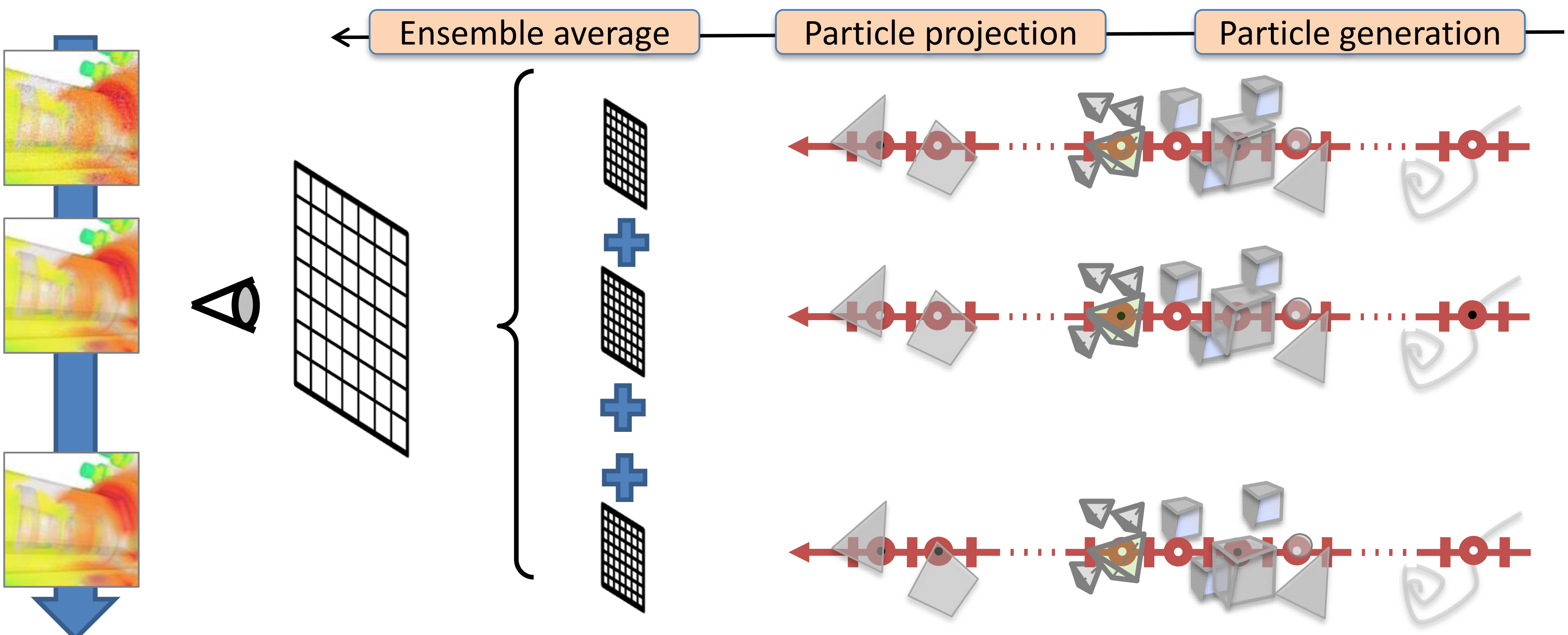
Particle-based volume rendering

K. Zhao et. al, "INTERACTIVE VISUALIZATION OF LARGE-SCALE 3D SCATTERED DATA FROM A TSUNAMI SIMULATION," International Journal of Industrial Engineering 24(2), p207-219.

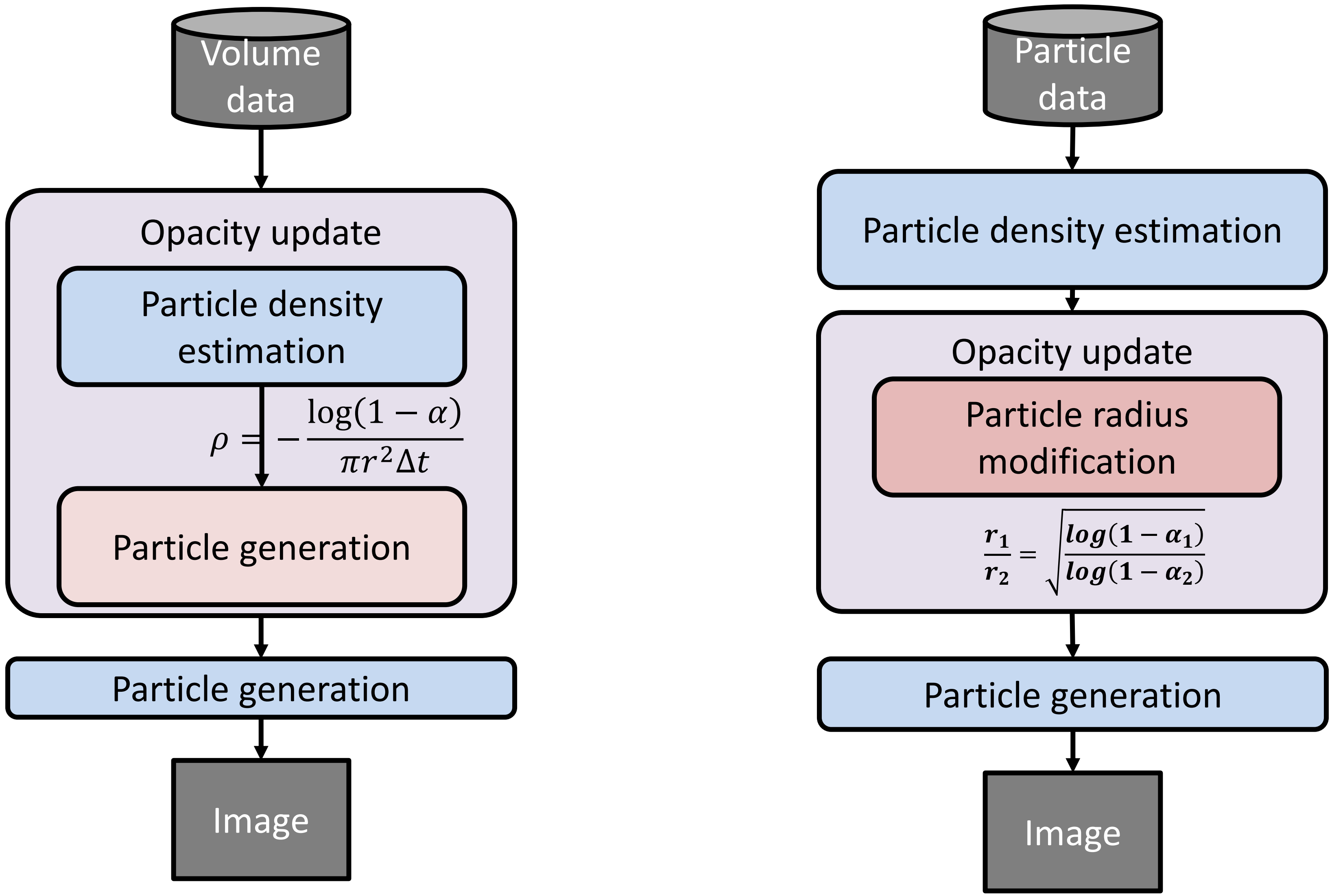


Particle-based volume rendering (PBVR)

- Generate a set of **opaque** particles
- Project the particles onto an image plane
- Use an ensemble average



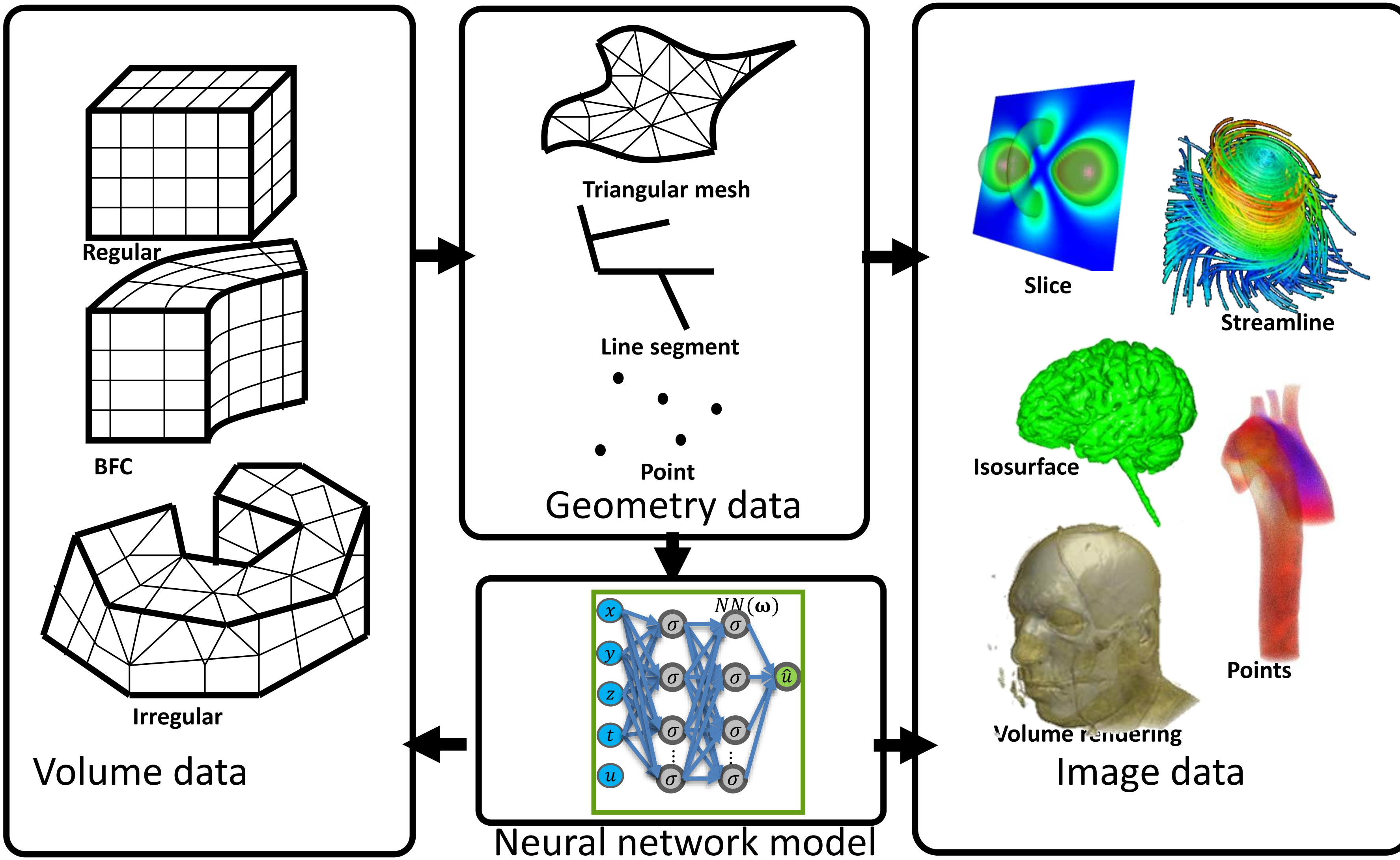
PBVR for point data



Visual data science research

AI-ENHANCED VISUALIZATION (AI4VIS)

AI-enhanced data visualization



Visual data science research

3-D BOOK DATA PAGE SEGMENTATION AND EXTRACTION

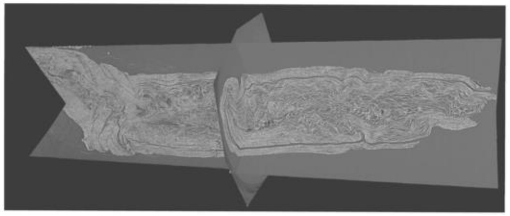
Introduction

Background

- In recent years, research cases using CT scanning equipment have been published regarding the decoding of ancient documents that cannot be opened.
- We are developing an analysis method for digitized literature, assuming that the literature is a booklet.

Attention: Historical literature

Need for non-invasive investigation



Damage caused by aging, fire, and flood damage in the literature

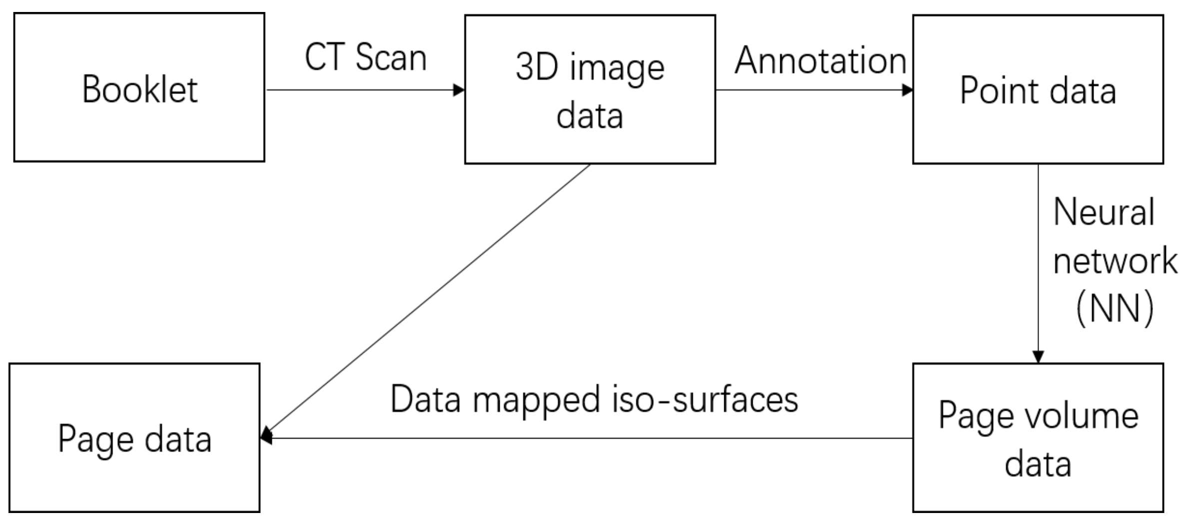
Research question: How can we extract page information from 3-D booklet data?

Proposed method: If we define a scalar field corresponding to the number of pages, generate an iso-surface, and map the booklet data on it, the question can be answered.

Proposed method

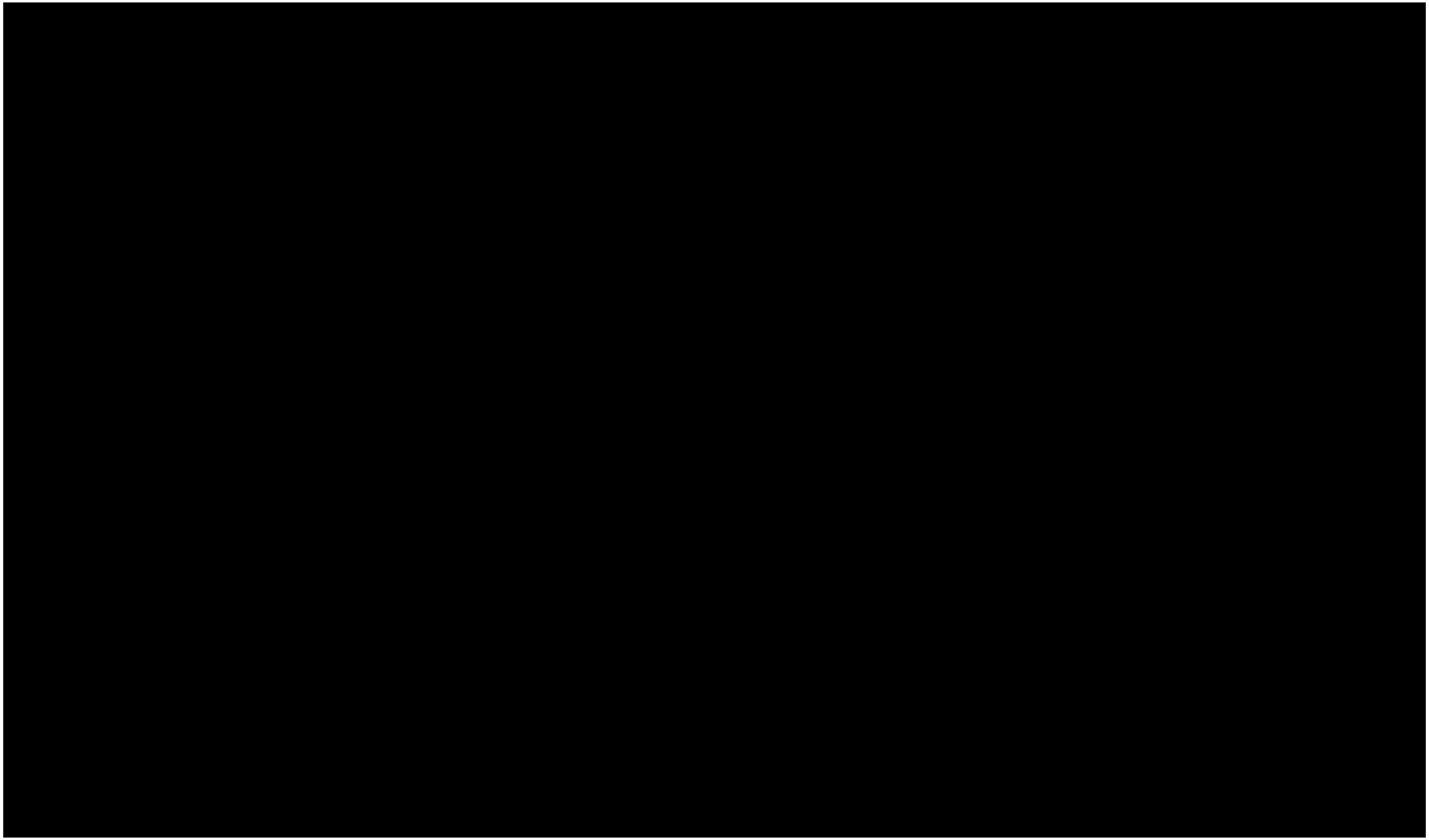
Generation of two volume datasets

- 3-D image data from scanned booklet
- Page volume data from annotated points in the 3-D image data



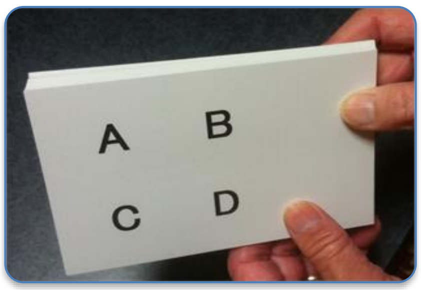
J. Ou, Z. Han, K. Koyamada, "Three-dimensional book data page segmentation and extraction method using Laplace equation,"Journal of Advanced Simulation in Science and Engineering, 8(2), pp. 223-236, 2021

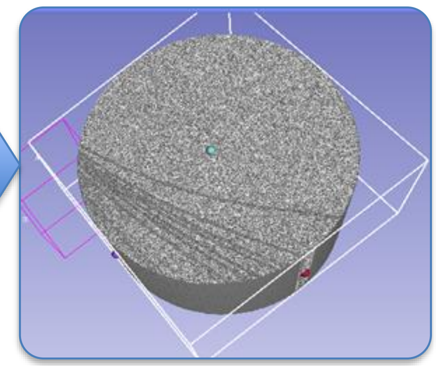
Overview



Digitalization of a booklet using 3-D CT

How can we digitalize a booklet?





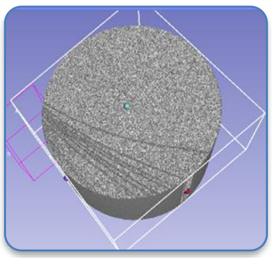
Booklet

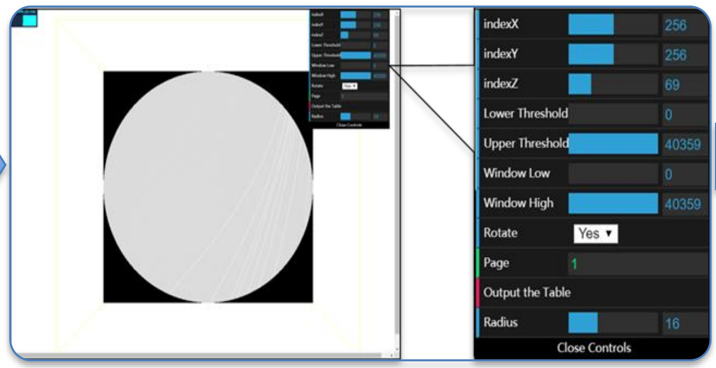
3-D CT

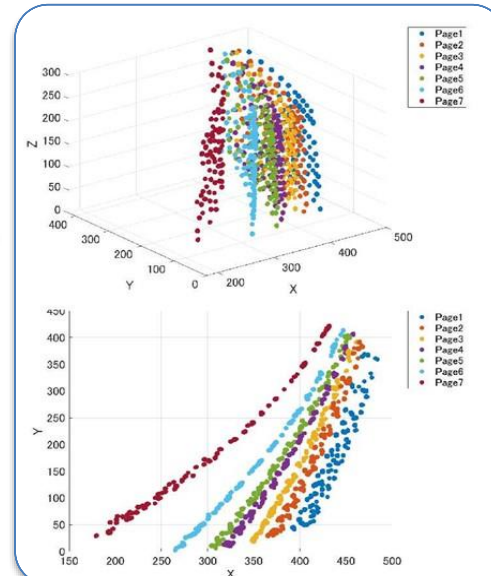
3-D image data

Annotation of 3-D image data

How can we annotate page numbers in a 3-D CT scanned document ?







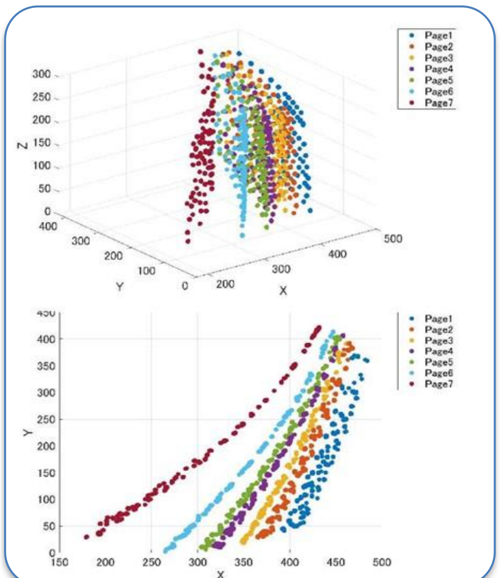
3-D image data

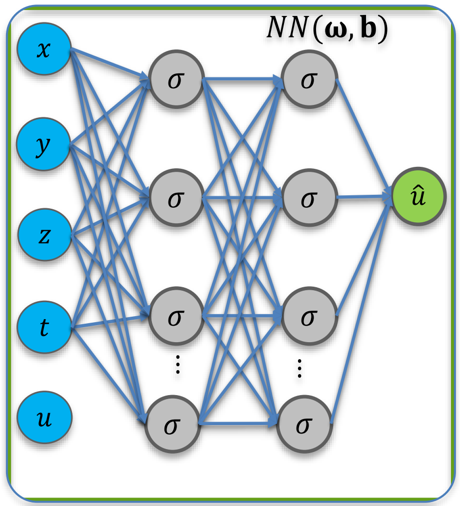
Annotation tool

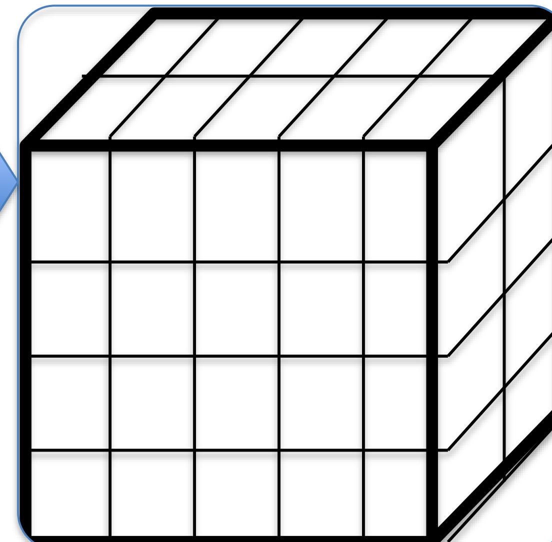
Point data

Generation of page volume data

How can we generate page volume data from Point data?







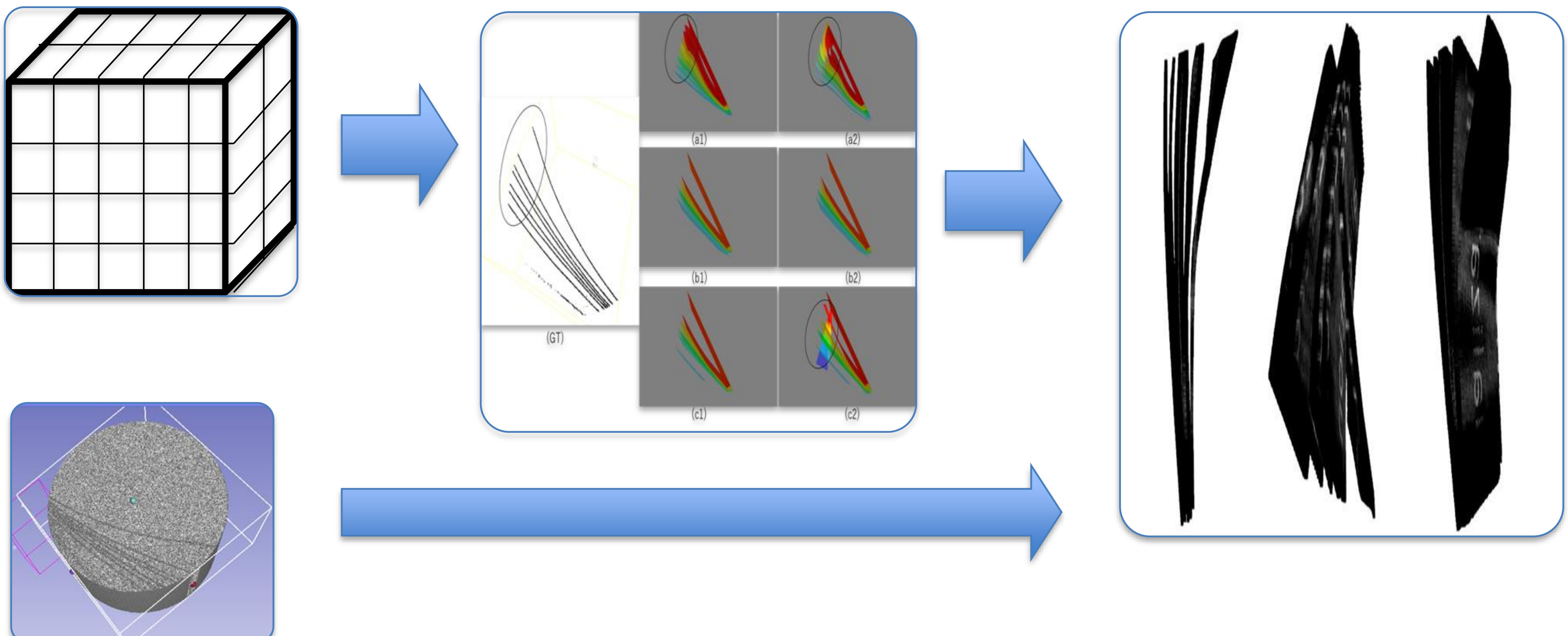
Point data

Neural network

Page volume data

Mapping 3-D image data onto page surface

How can we represent the page information on the page surface?



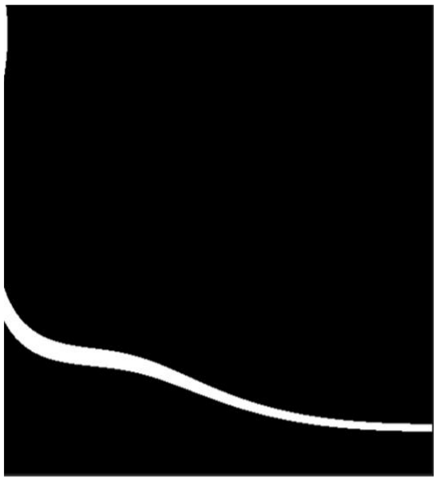
3-D image data

Page surface data

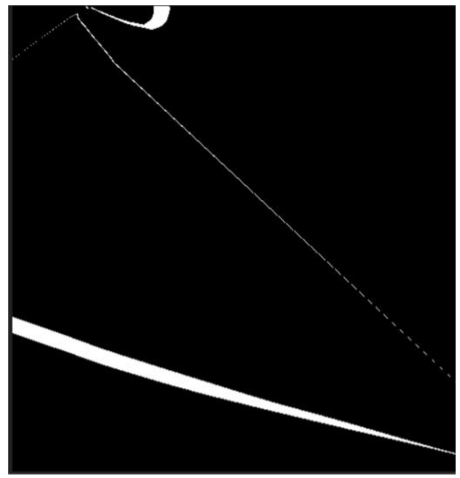
Printed page data

Results

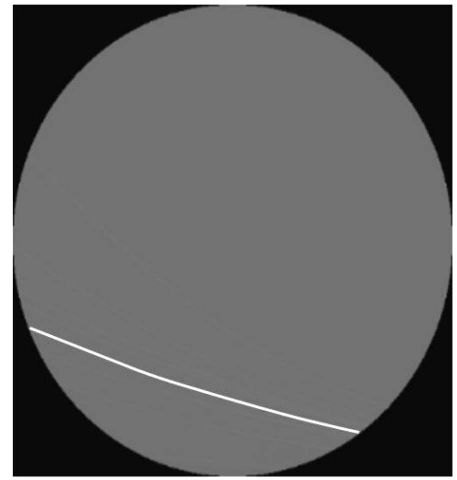
Comparison with Laplace's equation model



Page 3



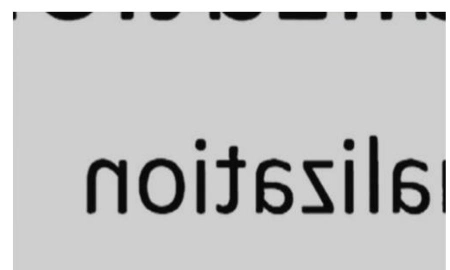
Page 3



Page 3 boundary in CT image (bright)

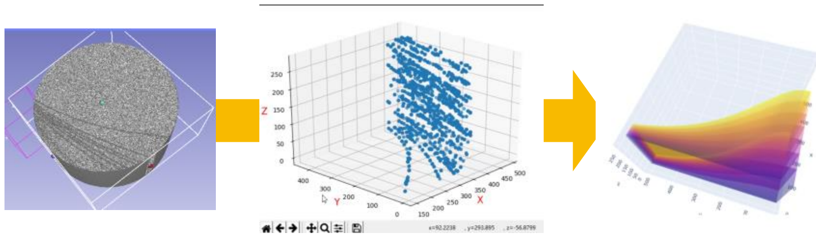
Laplace equation

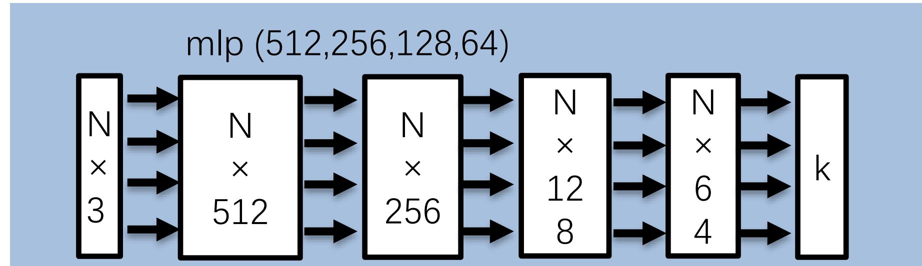
PointNet-based model



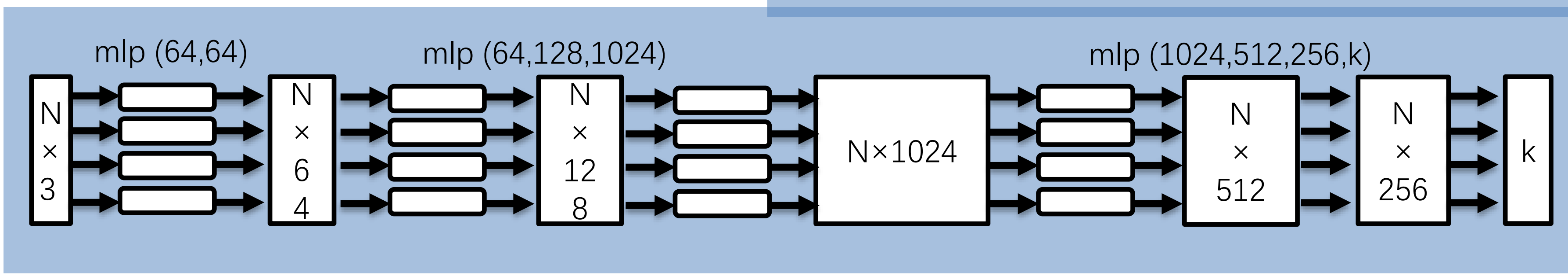
Text (ground trues)

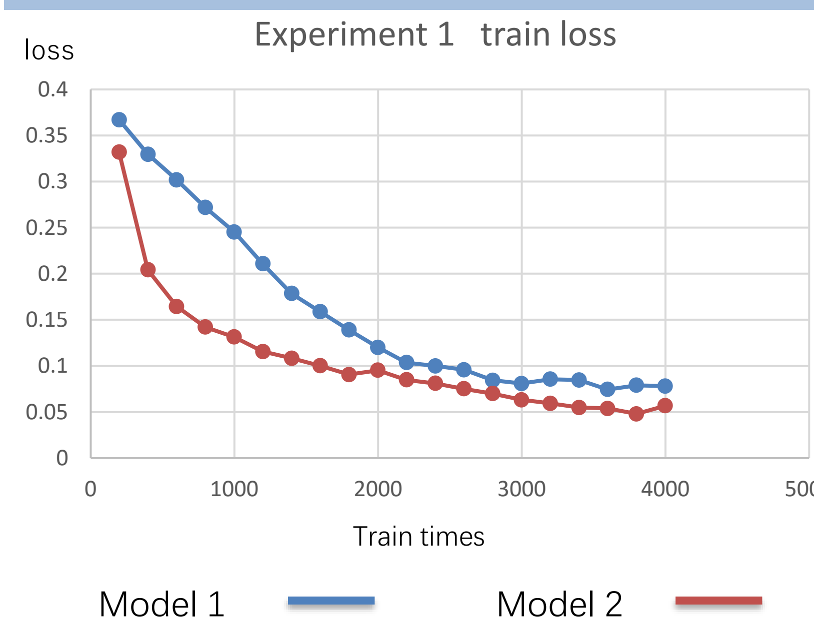
一般MLPモデル深層学習実験　（Model１）

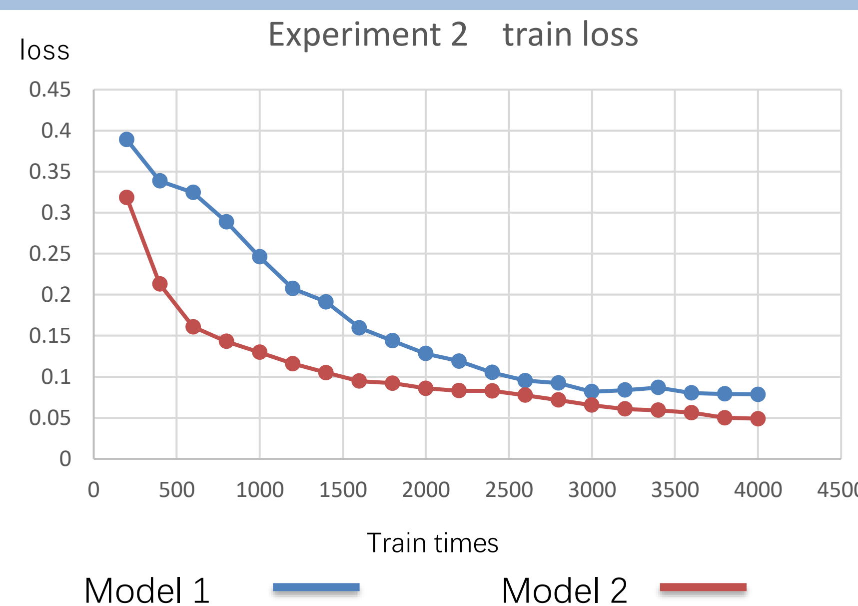




PointNetモデル深層学習実験　（モデル２）







モデル１より誤差が　38.79%　減った

International forum モデル１より誤差が　37.96%　減った

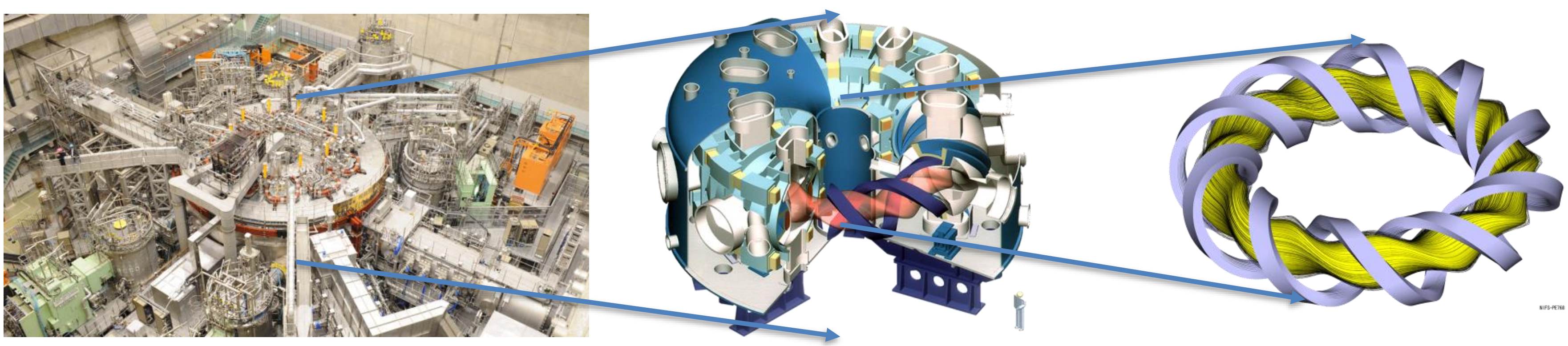
Visual data science research

VISUALIZATION OF PLASMA SHAPE IN THE LHD-TYPE HELICAL FUSION REACTOR

Large Helical Device（LHD）

Overview

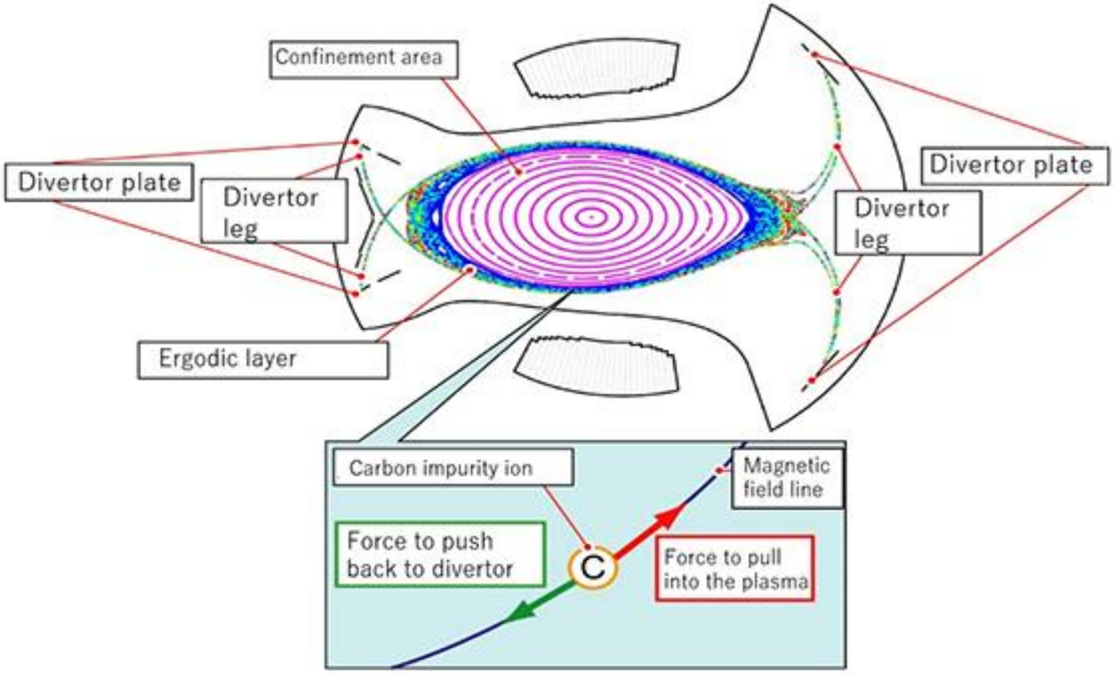
- The LHD is one of the world's largest helical devices with superconducting coils that can generate a strong magnetic field on a regular basis.
- In order to stably confine plasma in a magnetic field container (a basket of magnetic field lines), an infinitely circulating magnetic field line with no end is required to close an escape route for particles.
- In addition, in order to create a donut-shaped cage with these lines of magnetic force, it is necessary to add a twist to the lines of magnetic field.
- Heliotron configuration has excellent steady-state operation capability because the magnetic field configuration necessary for confining the plasma can be formed using only a pair of helical external coils.



Plasma regions

Cross section of LHD vacuum vessel and structure of magnetic field lines.

- The main structure is the "confinement region" where the plasma is confined by the cage of the magnetic field lines,
- The "ergodic region" around it, and the "divertor leg" that connects the layer and the "divertor plate". The divertor ejects impurities and the plate catches them.

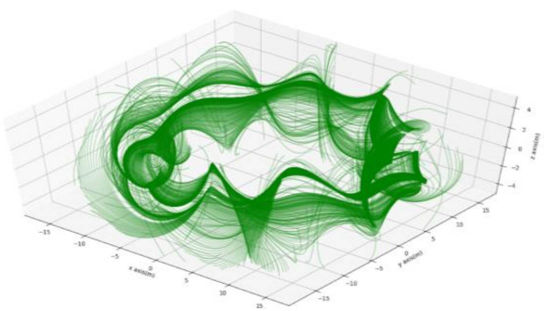


Introduction

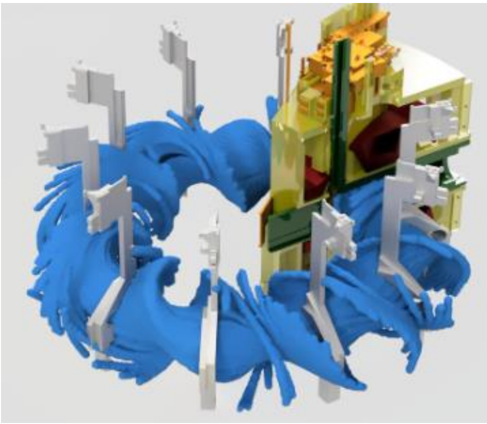
Background

- Visualization of the plasma region is indispensable for understanding the relationship with structures in the maintenance of fusion reactors.

Magnetic field lines in a fusion reactor



Relationship between fusion reactor structure and plasma region



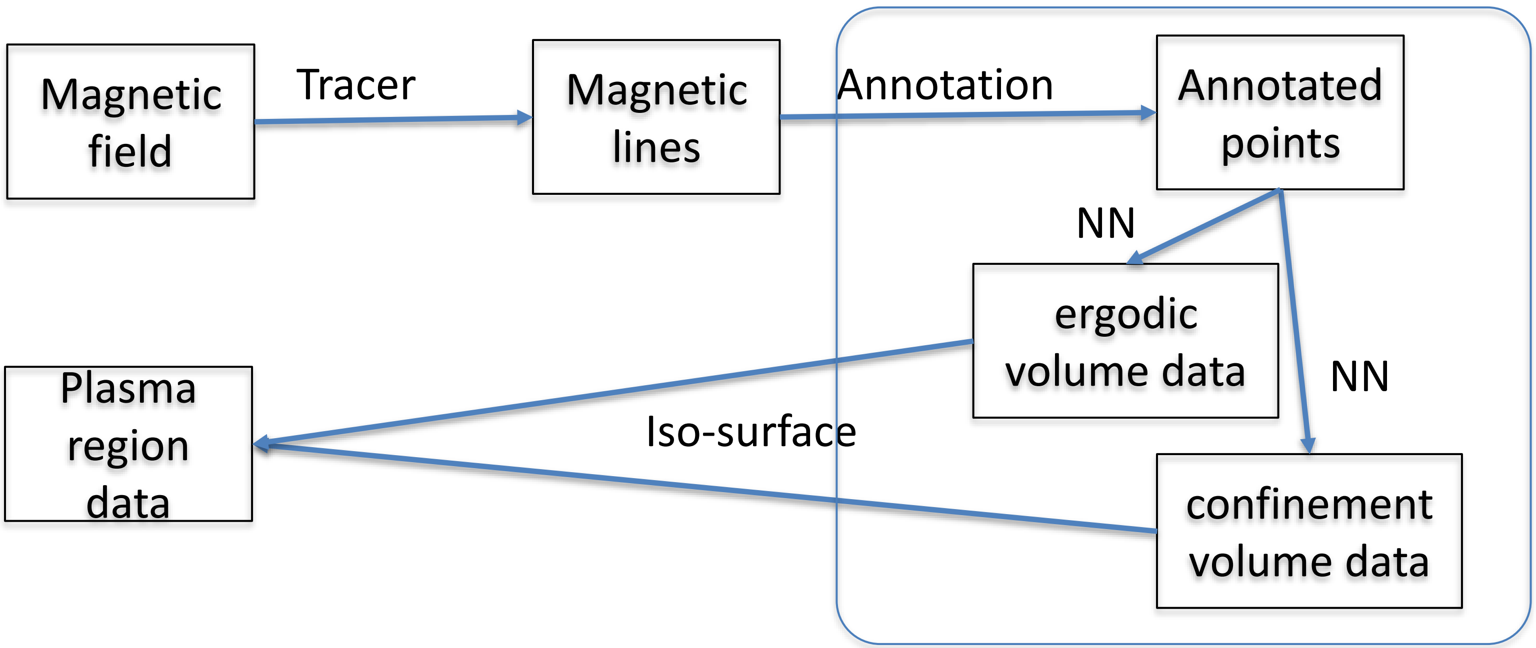
Research question: How can we derive a plasma region from a set of magnetic field lines?

Proposed method: If we define a scalar field from a set of magnetic field lines, the question can be answered.

Proposed method

Using NN model, generation of two volume datasets

- Confinement volume data
- Ergodic volume data



K. Hu, K. Koyamada, H. Ohtani, T. Goto, J. Miyazawa, Visualization of plasma shape in the lhd-type helical fusion reactor, ffhr, by a deep learning technique, Journal of Visualization 24 (6) (2021) 1141–1154

Overview

Inspecting the interference in a fusion reactor, FFHR, by an artificial neural network

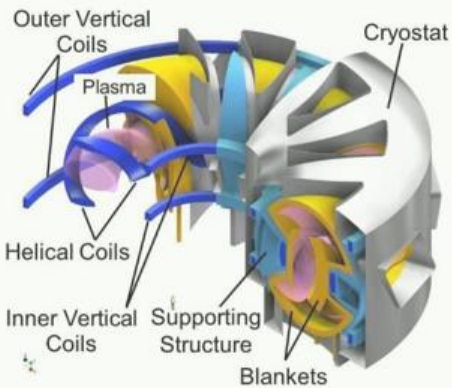


Figure 1: Schematic of the FFHR-d1 series fusion reactor.

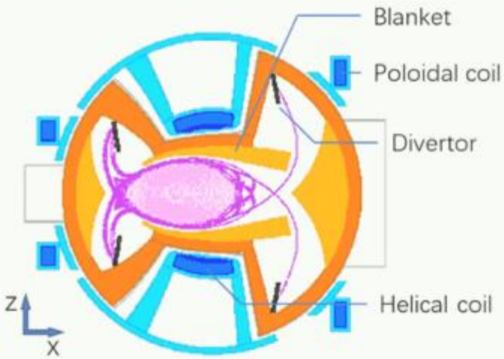


Figure 2: Poloidal cross-section of the FFHR at horizontally elongated plasma cross-section.

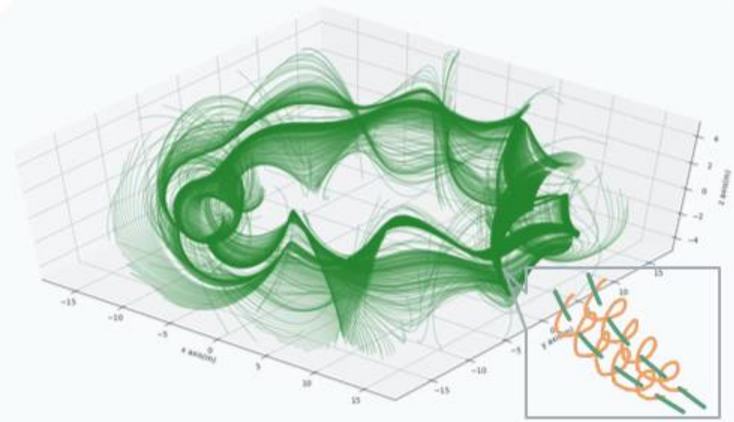


Figure 3: Rendering of all the magnetic field lines with transparence. In the grey frame, it shows two magnetic field lines with their Larmor radius.

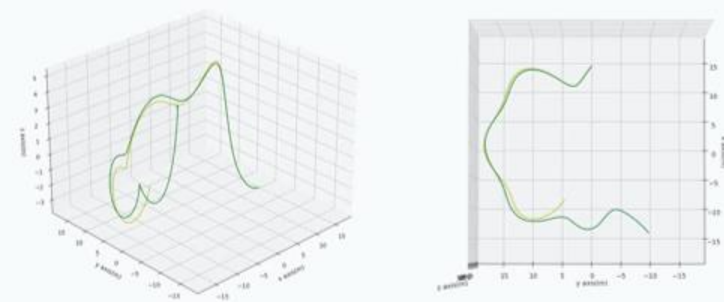
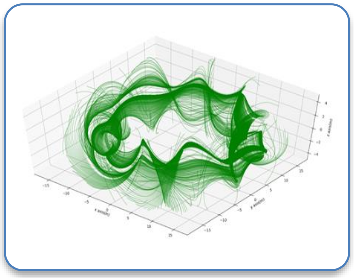
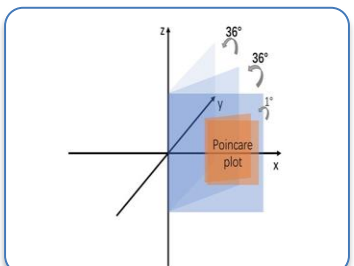


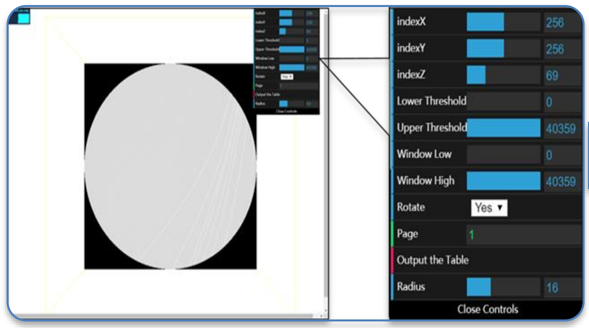
Figure 4: Two magnetic field lines in 3D space with oblique view and top view.

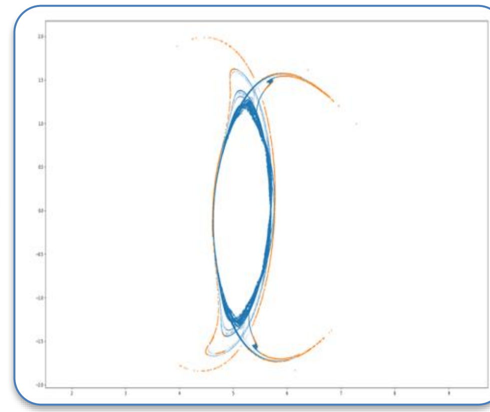
Annotation of magnetic lines

How caw we extract page information from a 3-D CT scanned document ?









Magnetic lines

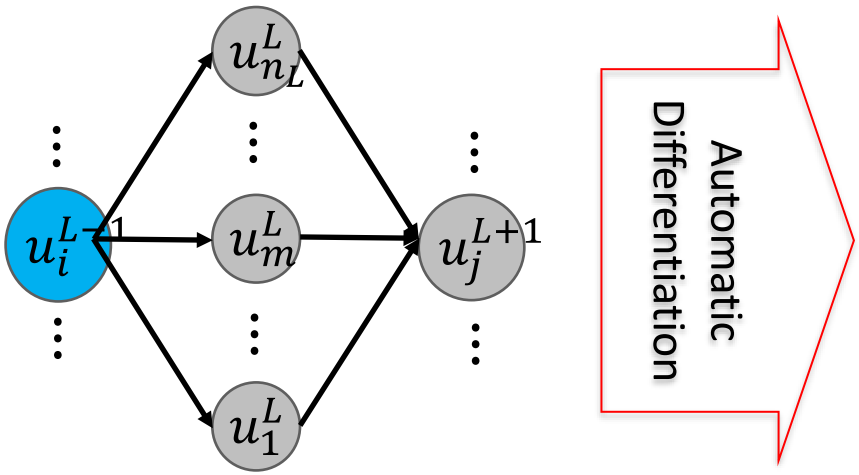
Annotation tool

Point data

Automatic Differentiation(AD)

AD is a set of techniques to evaluate the derivative of a function specified by a computer program.

The partial differential term u_j^{L+1} in the (L + 1)-th layer by u_i^{L-1} in the (L-1)-th layer can be calculated using the chain rule as follows.



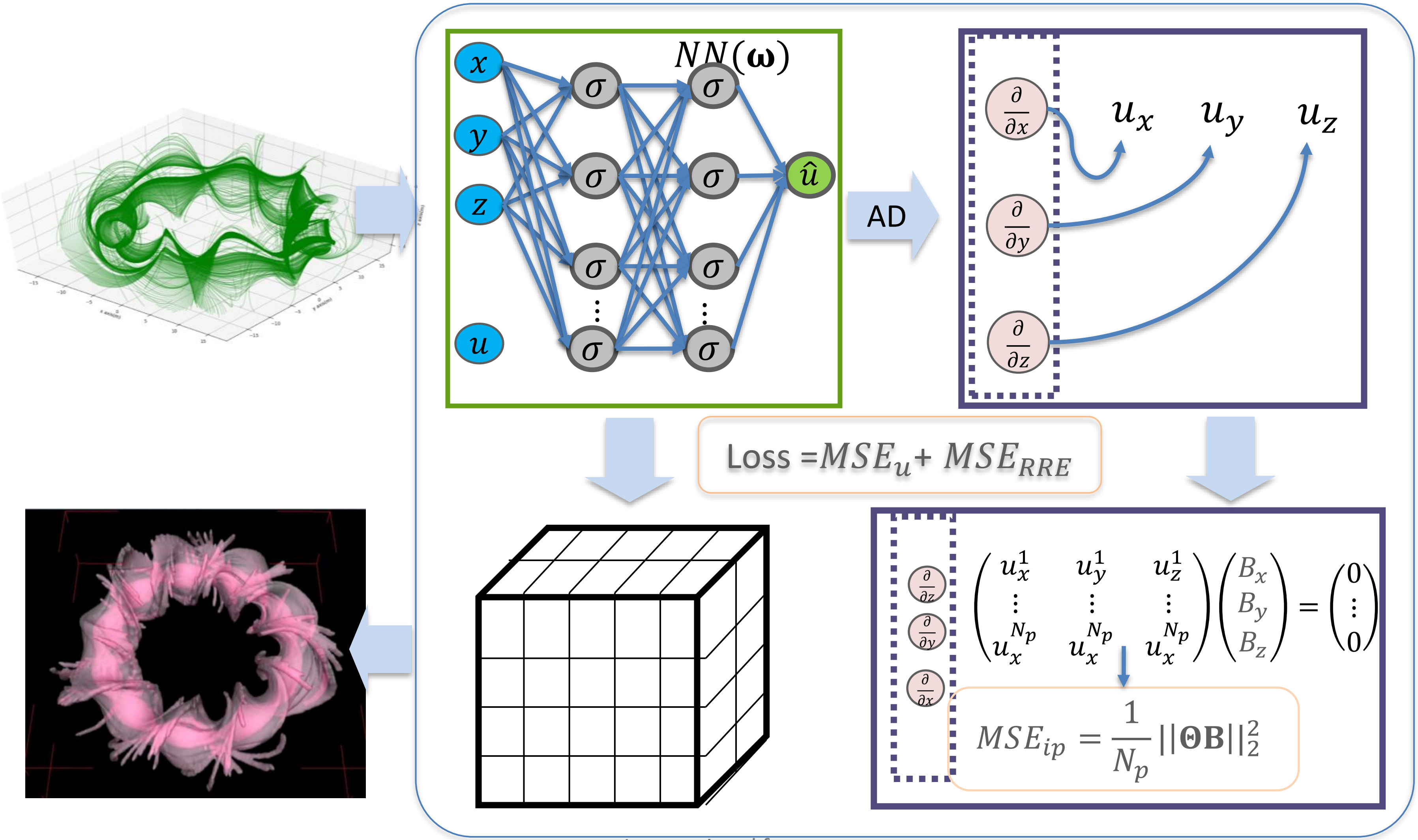
$$\frac{\partial u_j^{L+1}}{\partial u_i^{L-1}} = \sum_m^{n_L} \frac{\partial u_j^{L+1}}{\partial u_m^L} \frac{\partial u_m^L}{\partial u_i^{L-1}}$$

In the NN model, if this relation is applied such that the layers propagate back using the chain rule, the first-order term $\frac{\partial u_j^{L+1}}{\partial u_i^0}$ can be calculated. Here, u_i^0 represents spatiotemporal coordinates $(t, \boldsymbol{x})$.

$$\frac{\partial u_j^{L+1}}{\partial u_i^0} = \left(\frac{\partial u}{\partial x}\right) = \sum_m^{n_1}\left(\sum_m^{n_2}\left(\sum_m^{n_3}\left(\cdots\left(\sum_m^{n_L} \frac{\partial u_j^{L+1}}{\partial u_m^L} \frac{\partial u_m^L}{\partial u_i^{L-1}}\right)\cdots\right)\frac{\partial u_m^3}{\partial u_i^2}\right)\frac{\partial u_m^2}{\partial u_i^1}\right)\frac{\partial u_m^1}{\partial u_i^0}$$

Proposed technique

Add an inner product of gradient and magnetic field line direction term to the loss function

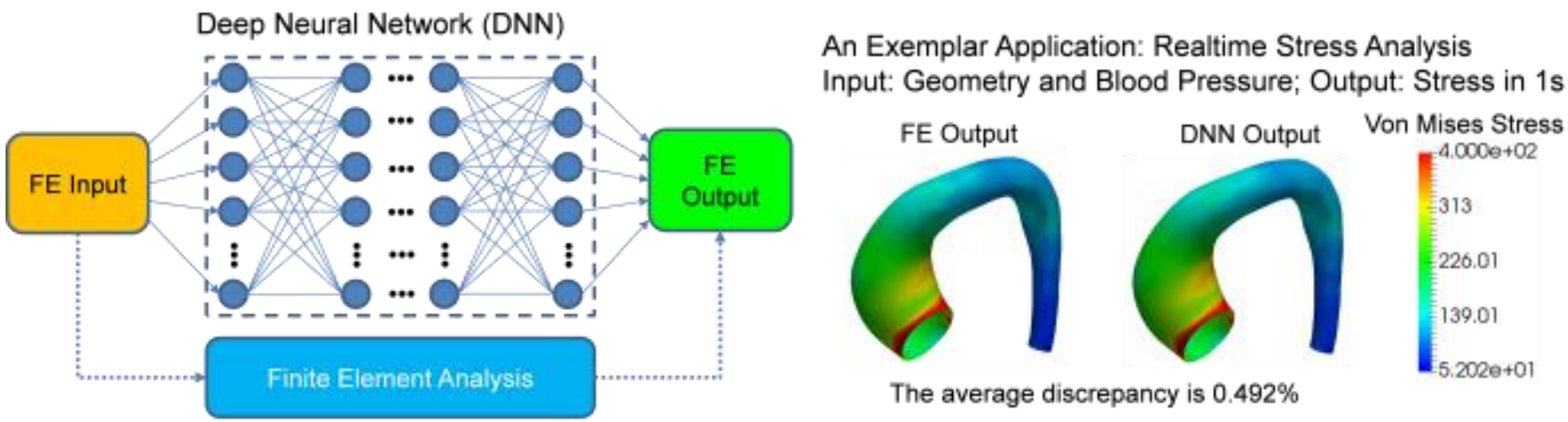


Visual data science research

SURROGATE MODEL

Engineering simulation

1. Engineering method has used the surrogate model when an outcome of interest cannot be easily directly measured, so a model of the outcome is used instead
2. Most engineering design problems require experiments and/or simulations to evaluate design objective and constraint functions as a function of design variables.
3. Surrogate model requires a diversity of input, including boundary conditions



Liang Liang, Minliang Liu, Caitlin Martin, and Wei Sun. A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite element analysis. *Journal of The Royal Society Interface*, 2018.

Compact model

A. Okamoto, et. al, 2020

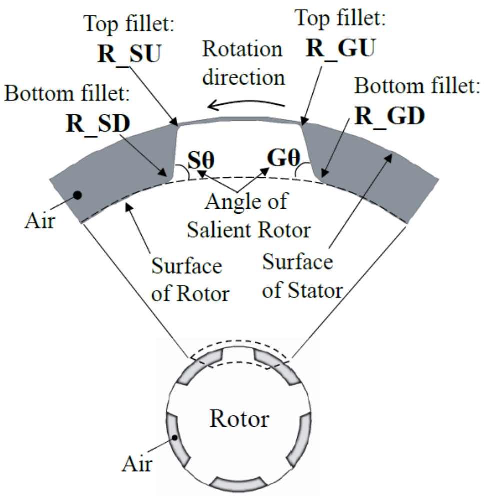
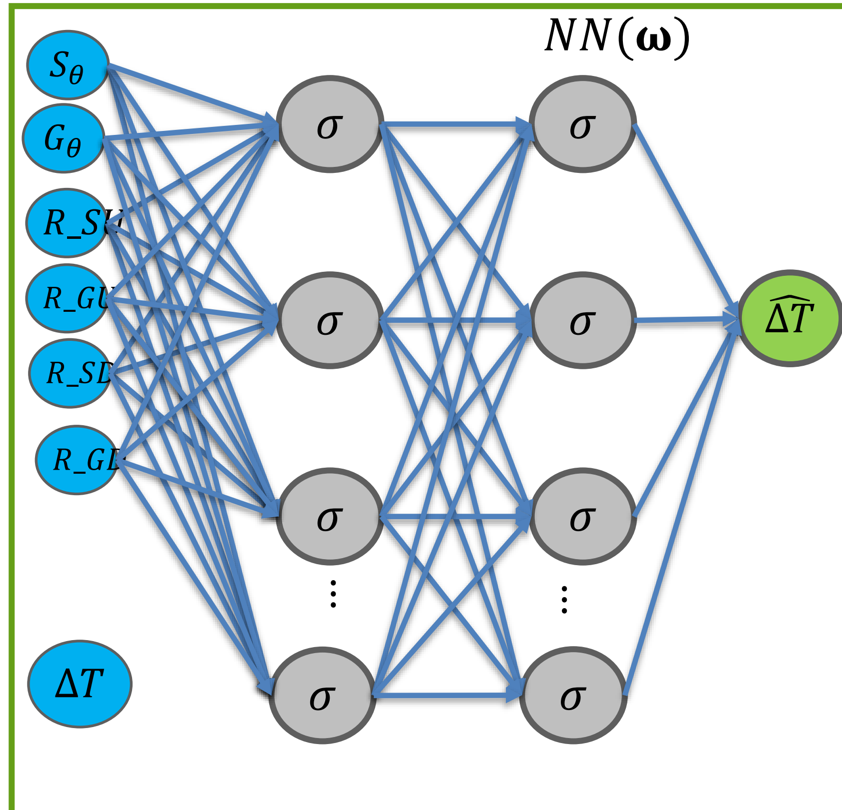


Fig.1 Analysis model

Table 1 Parameter of optimization

		Min	Max	Step size (Number of levels)
Angle of Salient Rotor[°]	Sθ	2	160	2 (80)
	Gθ			
Top fillet [mm]	R_SU	0.5	8	0.5 (16)
	R_GU			
Bottom Fillet [mm]	R_SD			
	R_GD			



Features of surrogate model

- Instantly predict analysis results by machine learning instead of numerical simulation(NS)
- Realizing an environment where you can enjoy the power of NS at the design
- Free from resource shortage by ultra-high-speed NN calculation
- Keep in mind **the physical consideration** of the prediction results
- **The validity of the surrogate model** is especially important

Reference to Partial Differential Equation(PDE) is essential for physical validity

Partial Differential Equation(PDE)

A differential equation

- in which an unknown function is a function of two or more variables
- which includes partial differential coefficients related to these of unknown functions

is called a PDE.

PDEs are often used in the field of natural science to describe natural phenomena related to fields such as fluids, gravitational fields, and electromagnetic fields.

Partial differential terms can be calculated by AD

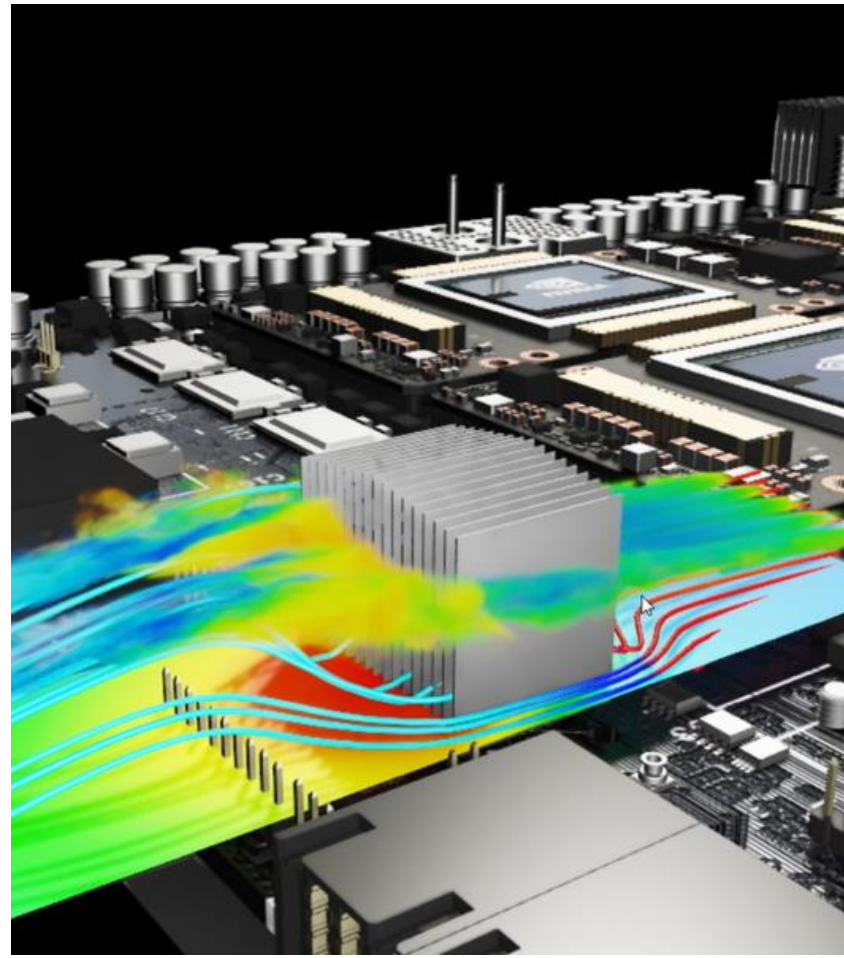
↓

How can we refer to PDEs?

NVIDIA MODULUS (SIMNET)

https://developer.nvidia.com/blog/nvidia-announces-modulus-a-framework-for-developing-physics-ml-models-for-digital-twins/

- A Framework for Developing Physics Machine Learning Neural Network Models
- Features
  - Novel Neural Network Architecture
  - Design Space Exploration
  - Optimized for Multi-Physics Problems



Visual data science research

VISUALIZATION-ENHANCED AI (VIS4AI)

Visual data science research

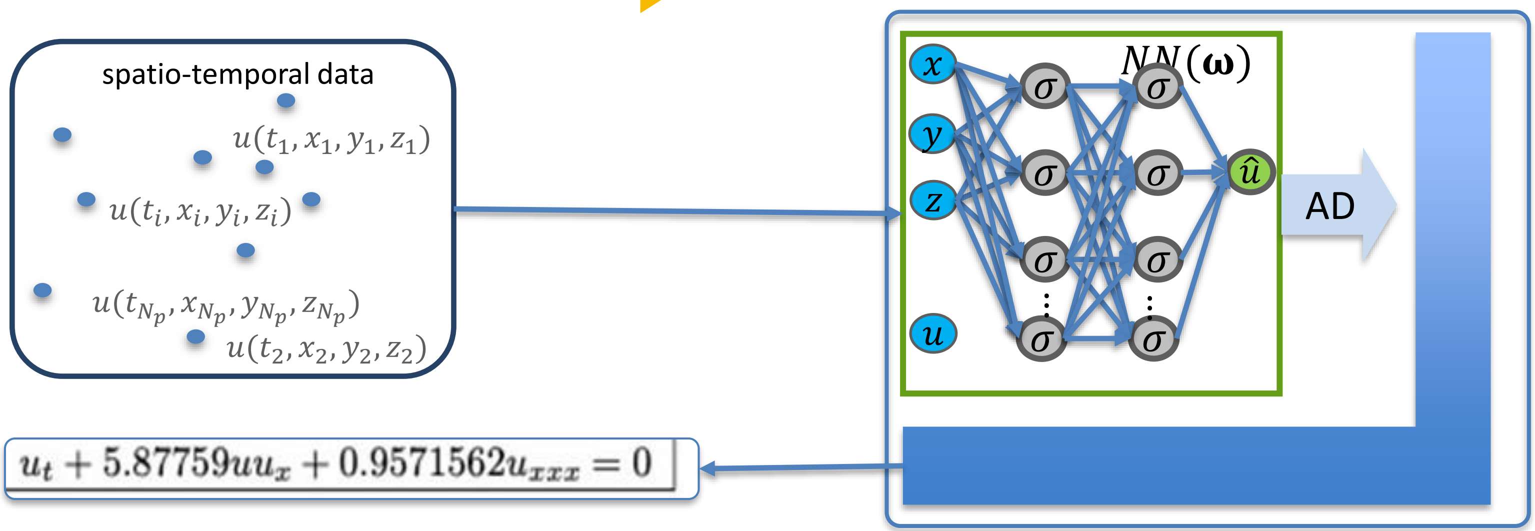
PDE DERIVATION FROM SPATIO-TEMPORAL DATA

PDE derivation

Background

- Explanatory models that use existing PDEs are very important for the utilization of big data from a variety of new phenomena, including new corona infections.

Measurement data → Partial differential equations governing the data



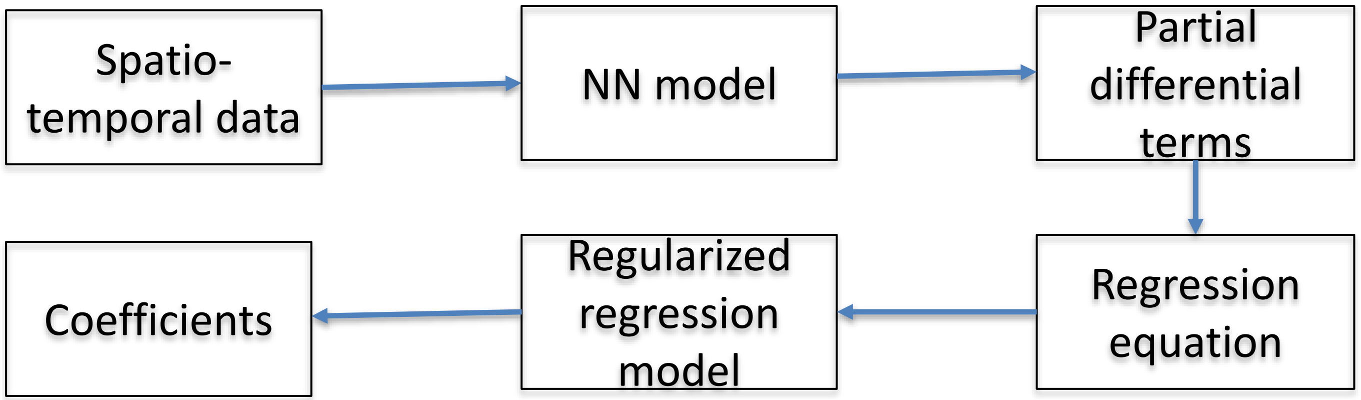
Research question: Can we derive a PDE from discrete spatio-temporal data?

Proposed method: If we add a regularized regression error term to the loss function, the question can be answered.

Proposed method

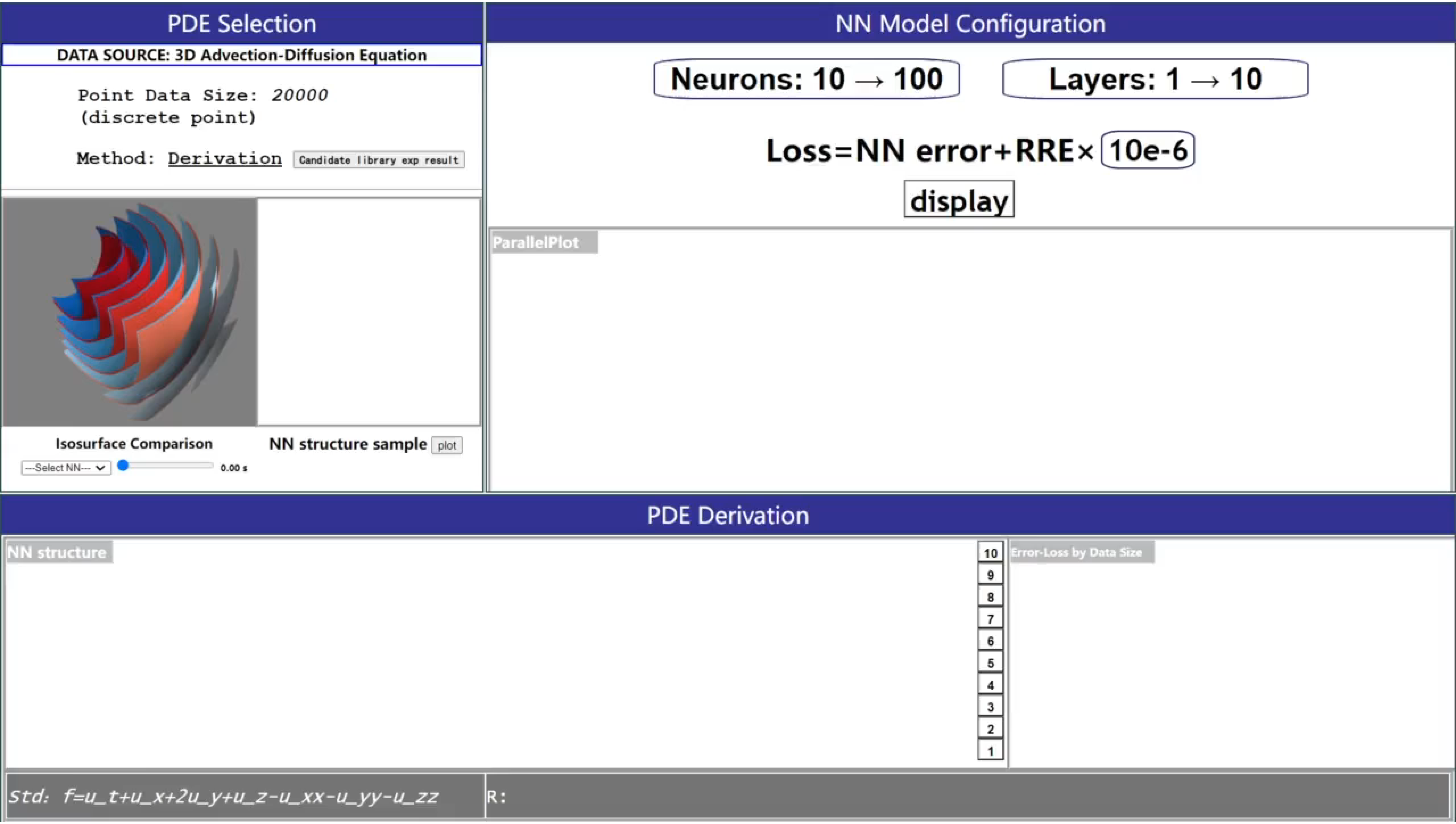
Derivation of partial differential equations(PDEs) from spatio-temporal data

- Pseudo measurement data from exact/FDM/FEM solution of PDE
- Partial differential terms from predefined library
- Coefficients determined using regularized regression analysis



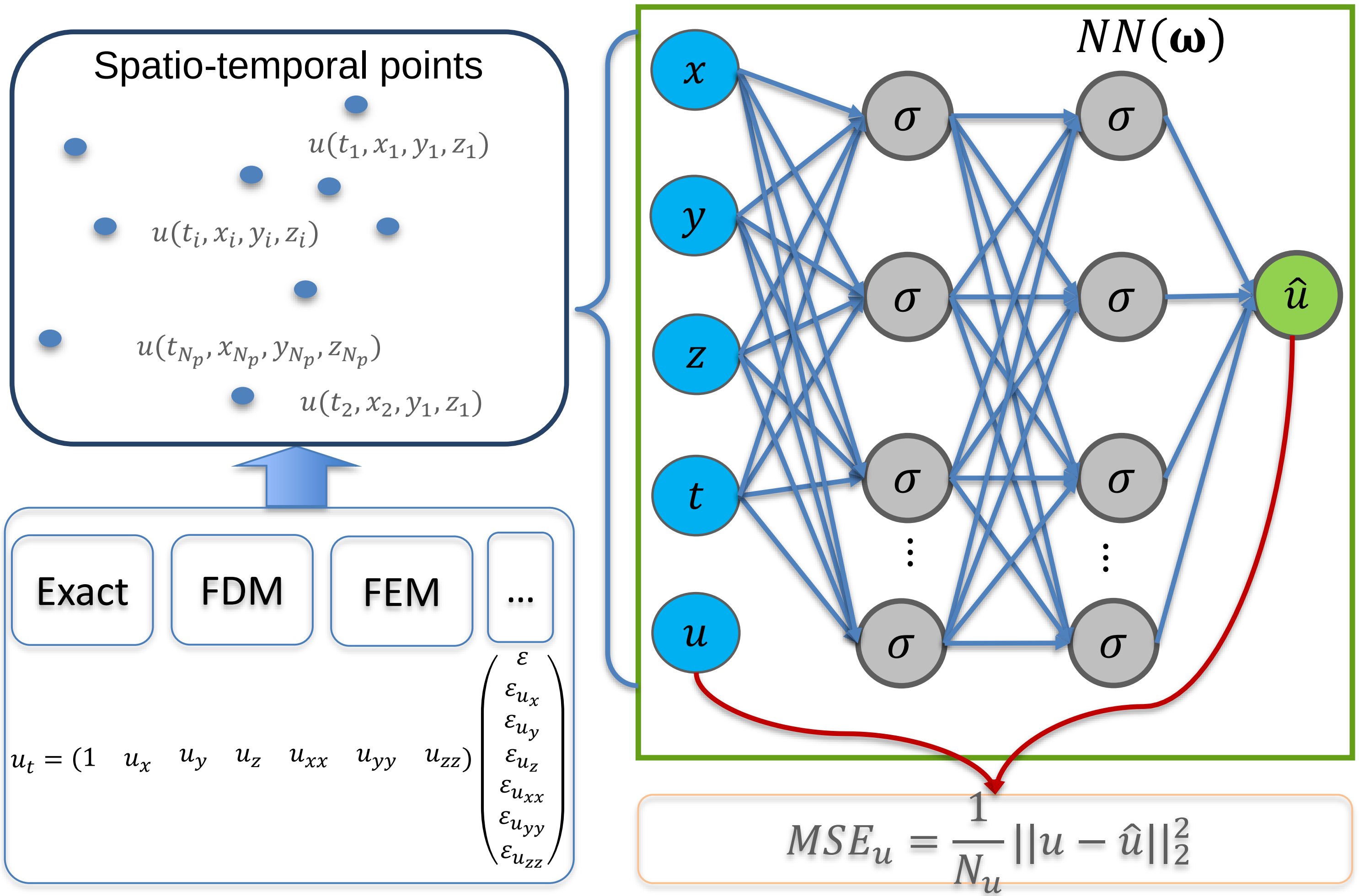
K.Koyamada, et al.,"Data-driven derivation of partial differential equations using neural network model," IJMSSC, 12/2, 2021

Overview



PDE Selection
DATA SOURCE: 3D Advection-Diffusion Equation
Point Data Size: 20000
(discrete point)
Method: Derivation
Candidate library exp result
Isosurface Comparison
---Select NN---
0.00 s
NN structure sample
plot
NN Model Configuration
Neurons: 10 → 100
Layers: 1 → 10
Loss=NN error+RRE× 10e-6
display
ParallelPlot
PDE Derivation
NN structure
10
9
8
7
6
5
4
3
2
1
Error-Loss by Data Size
Std: f=u_t+u_x+2u_y+u_z-u_xx-u_yy-u_zz
R:

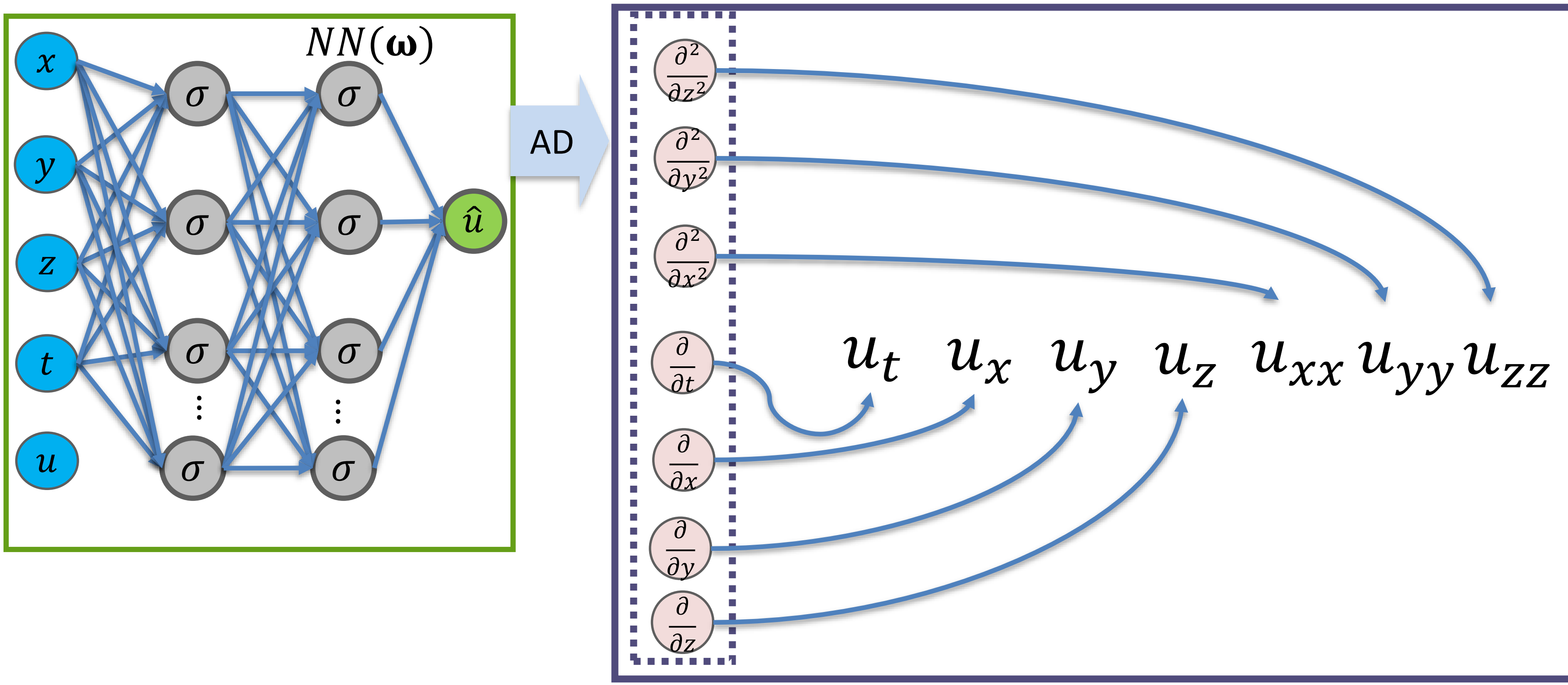
NN model from Spatio-temporal points



Spatio-temporal points

NN model

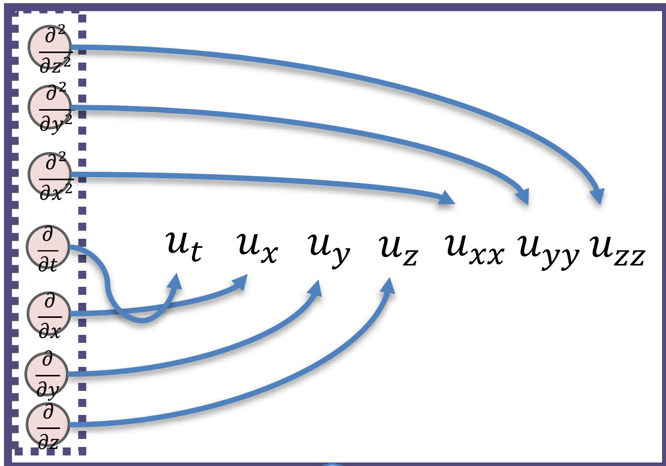
Partial differential terms from NN model



NN model

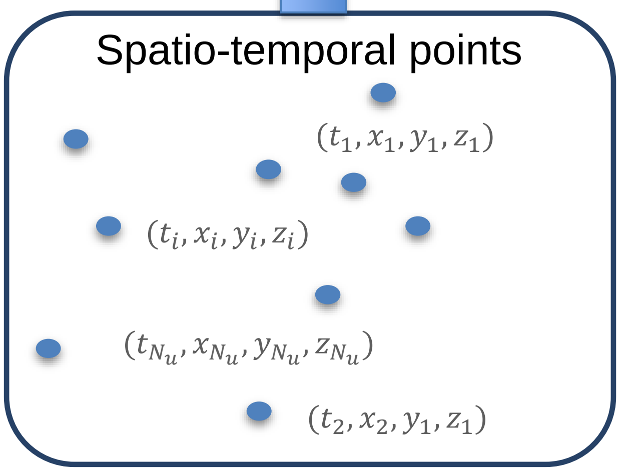
Partial differential terms

Regression equation from partial differential terms



$$\underbrace{\begin{pmatrix} u_t^1 \\ u_t^2 \\ u_t^3 \\ \vdots \\ u_t^{N_u-2} \\ u_t^{N_u-1} \\ u_t^{N_u} \end{pmatrix}}_{\mathbf{u}_t} = \underbrace{\begin{pmatrix} 1 & u_x^1 & u_y^1 & u_z^1 & u_{xx}^1 & u_{yy}^1 & u_{zz}^1 \\ 1 & u_x^2 & u_y^2 & u_z^2 & u_{xx}^2 & u_{yy}^2 & u_{zz}^2 \\ 1 & u_x^3 & u_y^3 & u_z^3 & u_{xx}^3 & u_{yy}^3 & u_{zz}^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & u_x^{N_u-2} & u_y^{N_u-2} & u_z^{N_u-2} & u_{xx}^{N_u-2} & u_{yy}^{N_u-2} & u_{zz}^{N_u-2} \\ 1 & u_x^{N_u-1} & u_y^{N_u-1} & u_z^{N_u-1} & u_{xx}^{N_u-1} & u_{yy}^{N_u-1} & u_{zz}^{N_u-1} \\ 1 & u_x^{N_u} & u_y^{N_u} & u_z^{N_u} & u_{xx}^{N_u} & u_{yy}^{N_u} & u_{zz}^{N_u} \end{pmatrix}}_{\boldsymbol{\Theta}} \underbrace{\begin{pmatrix} \varepsilon \\ \varepsilon_{u_x} \\ \varepsilon_{u_y} \\ \varepsilon_{u_z} \\ \varepsilon_{u_{xx}} \\ \varepsilon_{u_{yy}} \\ \varepsilon_{u_{zz}} \end{pmatrix}}_{\boldsymbol{\varepsilon}}$$

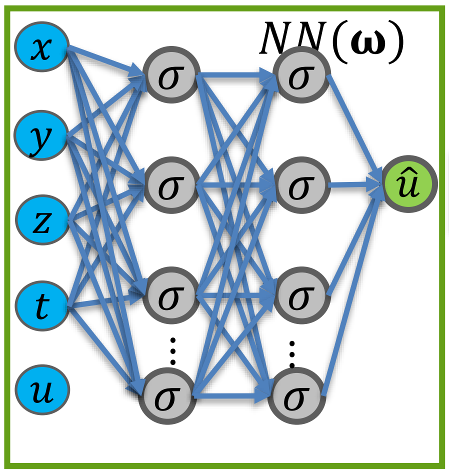
$$\mathbf{u}_t = \boldsymbol{\Theta} \quad \boldsymbol{\varepsilon}$$



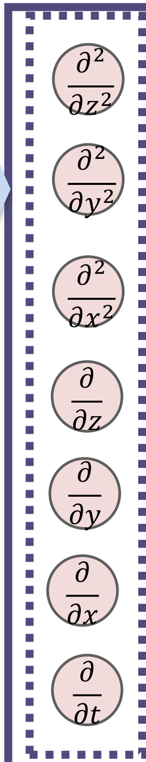
Partial differential terms

Regression equation

Regularized regression equation



AD



$$\mathbf{u}_t = \boldsymbol{\Theta}\boldsymbol{\varepsilon}$$

$$MSE_{RRE} = \frac{1}{N_u}||\mathbf{u_t} - \boldsymbol{\Theta}\boldsymbol{\varepsilon}||_2^2 + \lambda\frac{1}{N_{\boldsymbol{\varepsilon}}}||\boldsymbol{\varepsilon}||_2^2$$

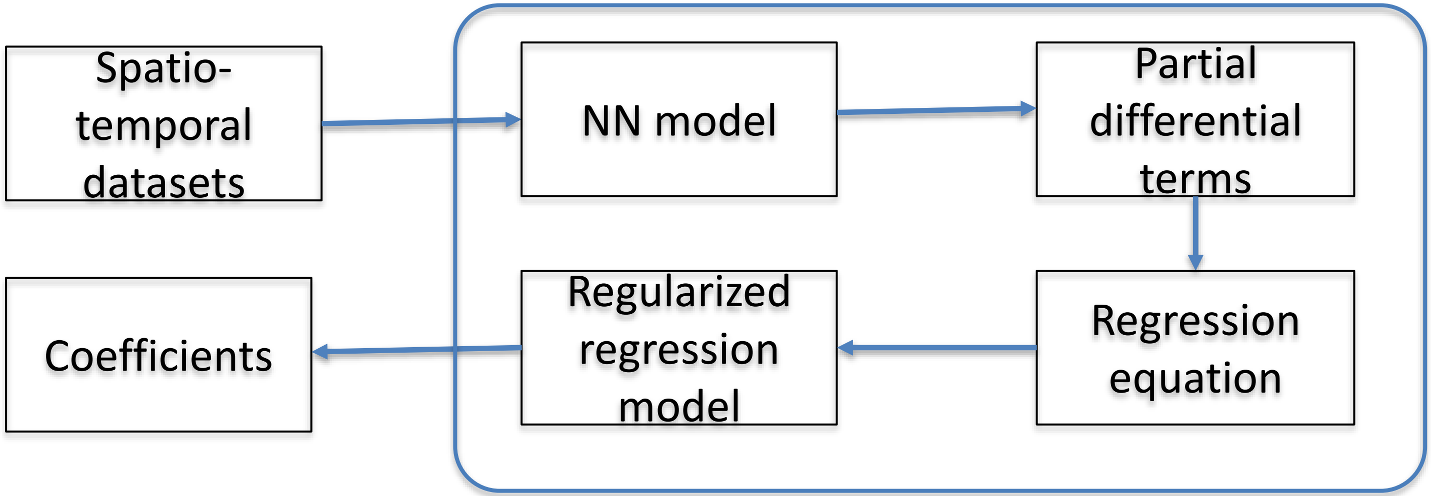
NN model

Regularized regression equation

Physics informed NN(PINN)

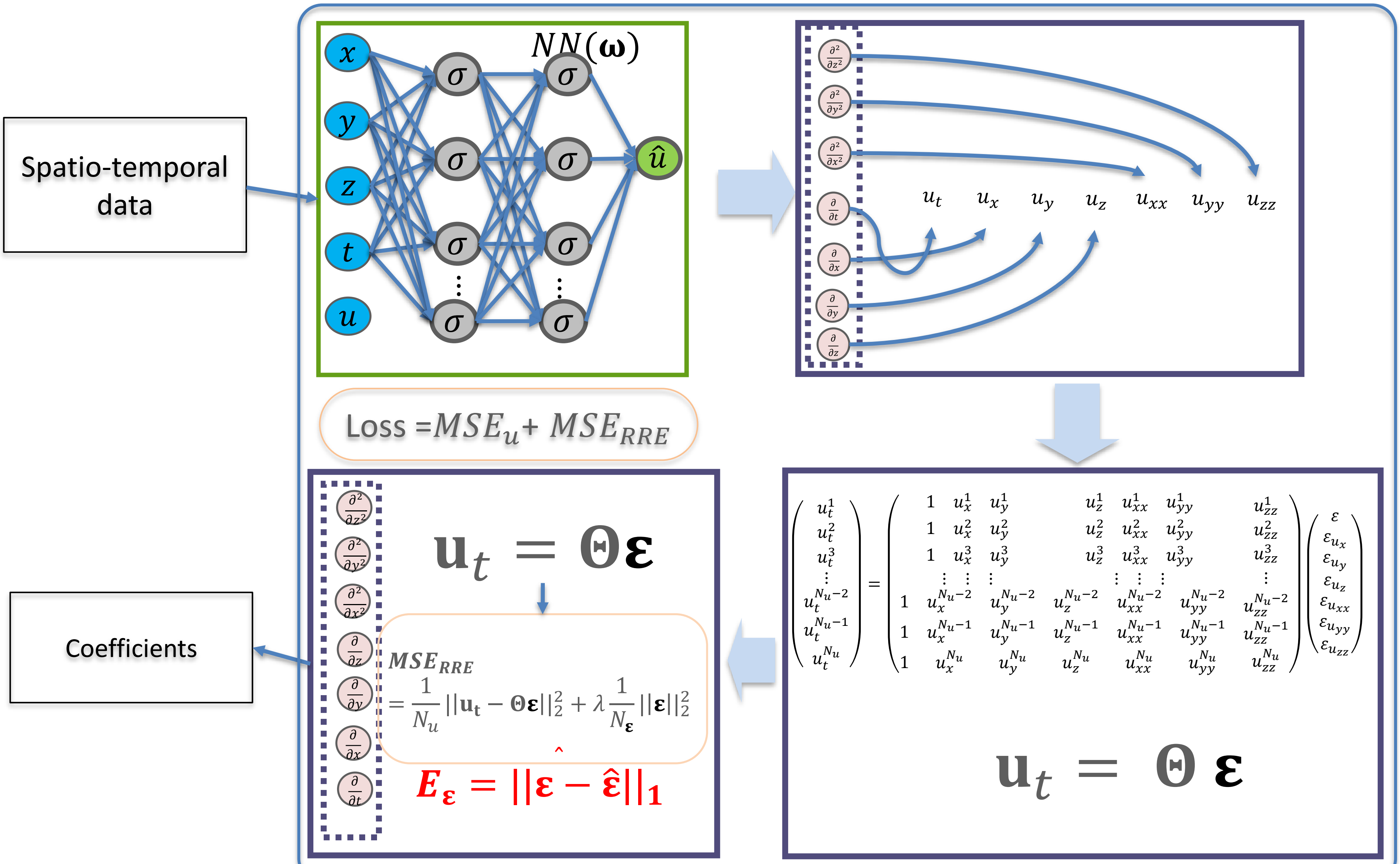
Add MSE_{RRE} to the loss function of NN model

- Adequate training of NN model
- Multi-objective optimization



M. Raissi et. al,"Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." Journal of Computational Physic, 378, 686-707, 2019

PDE derivation



Case study

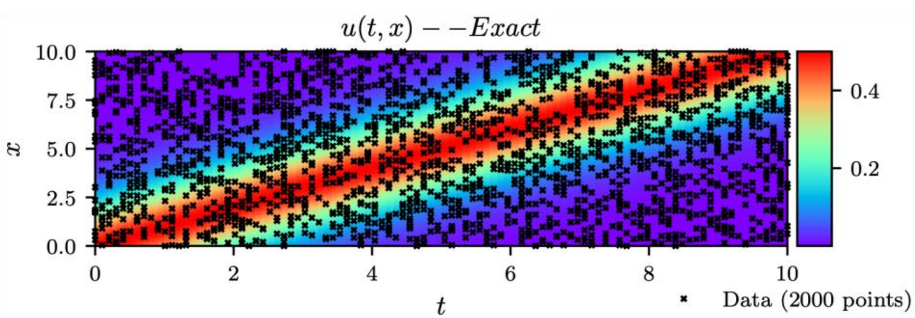
The training data set is generated from exact solutions of a PDE

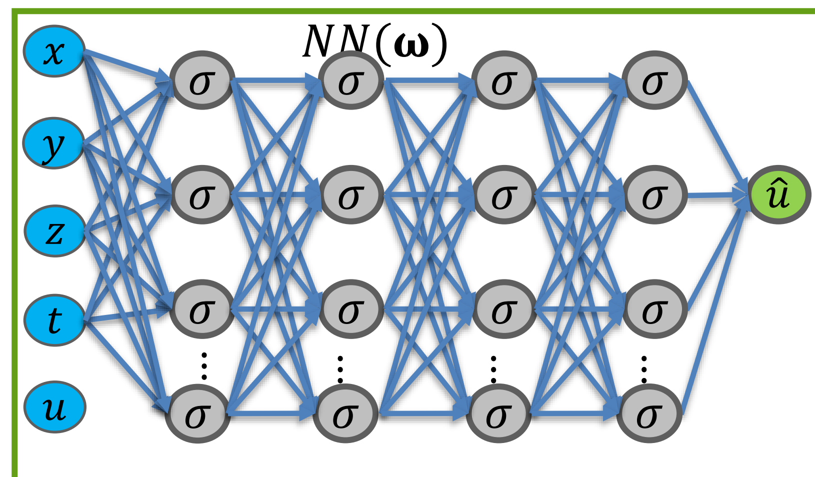
Random sampling is conducted inside a given spatio-temporal region

Uniform neural networks are employed

The PDE derivation errors are visualized in a parameter space.

- Number of layers(N_l): $1 \leq N_l \leq 5$
- Number of neurons(N_n): $1 \leq N_n \leq 50$





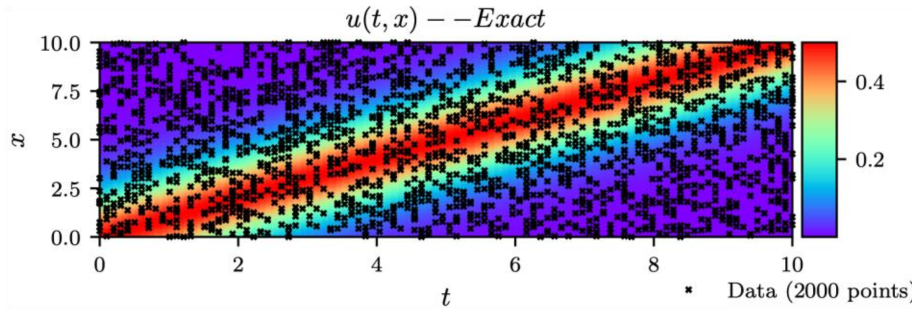
Case study: KdV equation

The training data set is generated from a specific exact solution that describes one initial condition.

$$u(0,t) = \frac{1}{2} sech^2(\frac{1}{2}(-t))$$

KdV equation is one of the nonlinear PDEs that describe the movement of waves in a shallow, constant channel and is often applied to the analysis of traffic flow.

$$\frac{\partial u}{\partial t} = -6u\frac{\partial u}{\partial x} - \delta\frac{\partial^3 u}{\partial x^3}$$

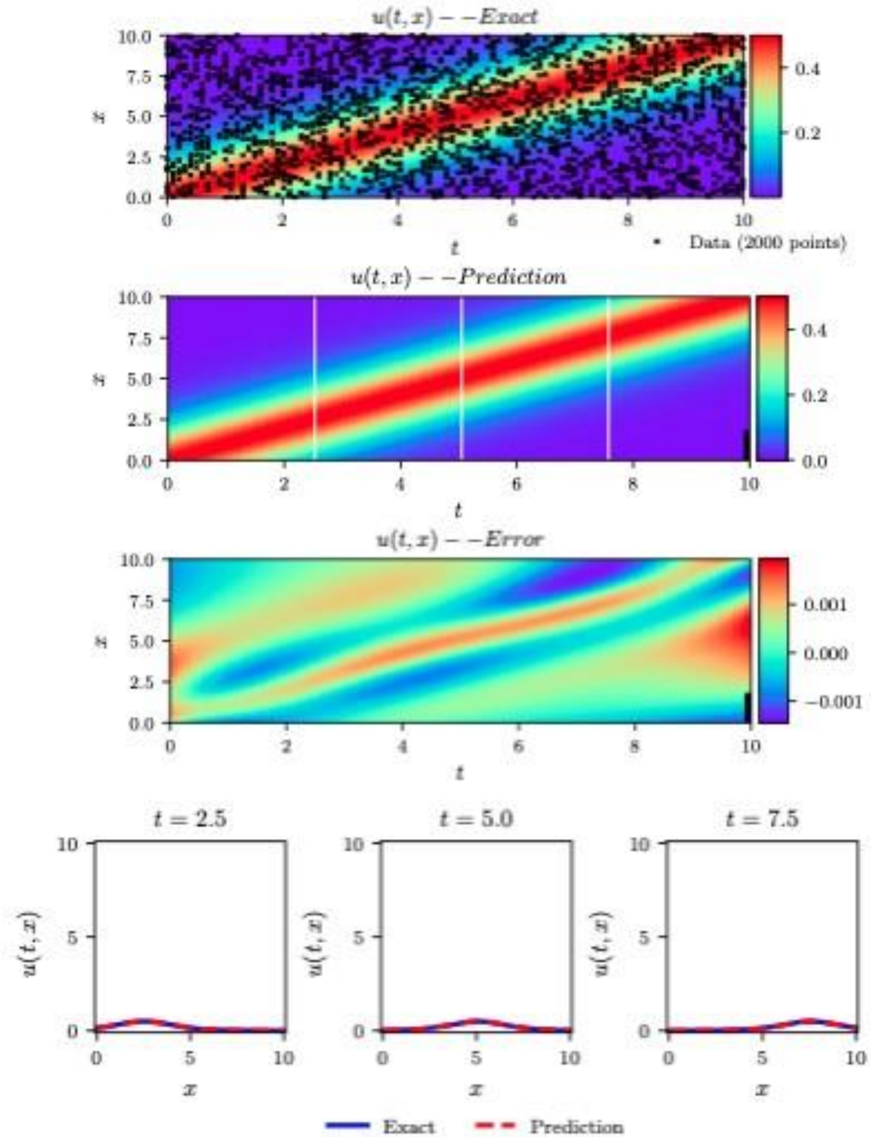


Exact solution with parameters set to δ = 1:

$$u(x,t) = \frac{1}{2} sech^2(\frac{1}{2}(x-t))$$

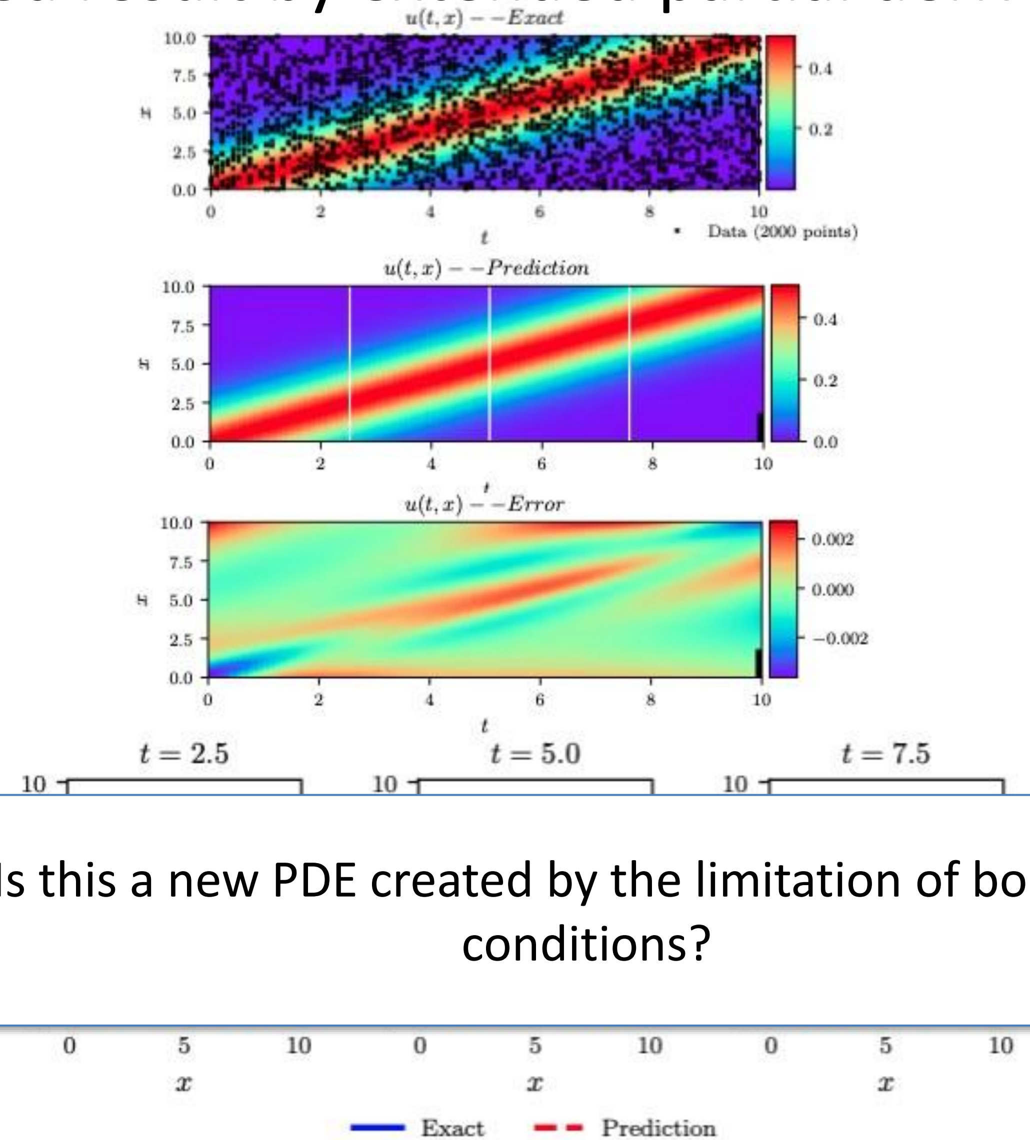
Rudy S H, Brunton S L, Proctor J L, et al. Data-driven discovery of partial differential equations[J]. Science Advances, 2017, 3(4): e1602614.

Derived results from the minimum partial derivative library



Correct PDE	$u_t + 6uu_x + 1u_{xxx} = 0$
Identified PDE (clean data)	$u_t + 5.87759uu_x + 0.9571562u_{xxx} = 0$

Derived result by extended partial derivative library



Is this a new PDE created by the limitation of boundary conditions?

Correct PDE	$u_t + 6uu_x + 1u_{xxx} = 0$
Identified PDE (clean data)	$u_t + 0.92477u_x + 0.40510uu_x + 0.28091u^2u_x + 0.0928930u_{xxx} = 0$

Hyper-parameter space

Error(NN model/PDE derivation) distribution in a hyper-parameter space

For uniform neural networks, the space is 2-D.

The PDE derivation errors $E_{\boldsymbol{\varepsilon}}(N_l, N_n)$ and NN model errors $MSE_{\mathbf{u}}(N_l, N_n)$are visualized in 2-D grids.

- Number of layers(N_l): $1 \leq N_l \leq 5$
- Number of neurons(N_n): $1 \leq N_n \leq 50$

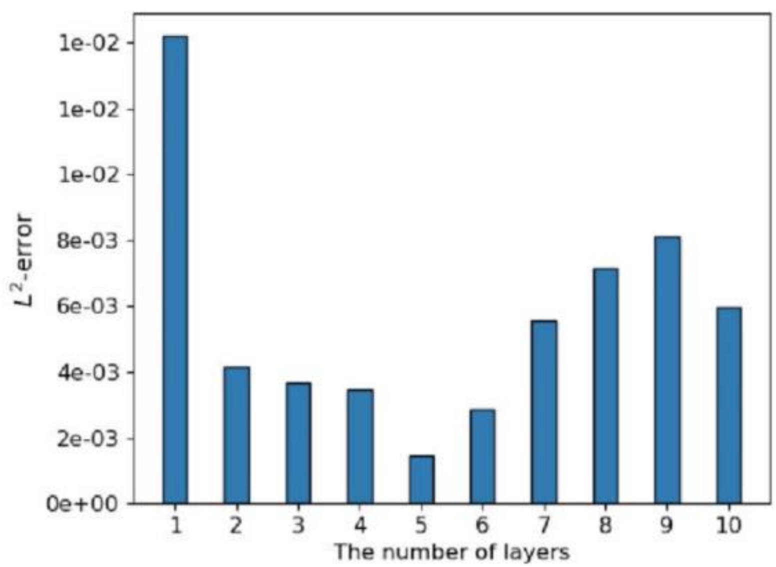
	1	2	3	4	5	6	7	8	9	…	47	48	49	50
1	0.769355	0.757662	0.277344	0.096872	0.194649	0.774735	0.442238	0.320686	0.867174	…	0.500614	0.062523	0.925974	0.671173
2	0.480603	0.969215	0.221868	0.166101	0.588281	0.30394	0.548226	0.503627	0.616705	…	0.610041	0.808881	0.460379	0.258919
3	0.213615	0.868969	0.348935	0.431339	0.918797	0.271171	0.113798	0.555384	0.534507	…	0.829246	0.840274	0.092983	0.224193
4	0.842262	0.775692	0.09756	0.399266	0.706456	0.206162	0.755654	0.637287	0.344111	…	0.367762	0.498399	0.578043	0.078828
5	0.006591	0.581007	0.538789	0.993953	0.320453	0.198827	0.233933	0.846745	0.050477	…	0.731686	0.617719	0.787064	0.064282

Error(NN model) distribution

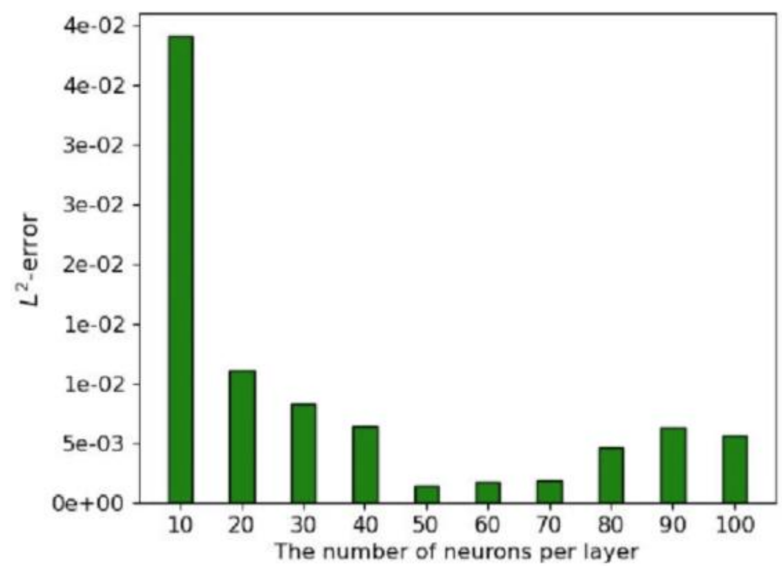
Chen, X., Chen, R., Wan, Q. *et al.* An improved data-free surrogate model for solving partial differential equations using deep neural networks. *Sci Rep* **11,** 19507 (2021).

Error(NN model) distribution in

- The number of layers): $1 \leq N_l \leq 10$
- The number of neurons: $10 \leq N_n \leq 100$



(a) L^2-error vs. layers (the number of neurons per layer is fixed at 50)



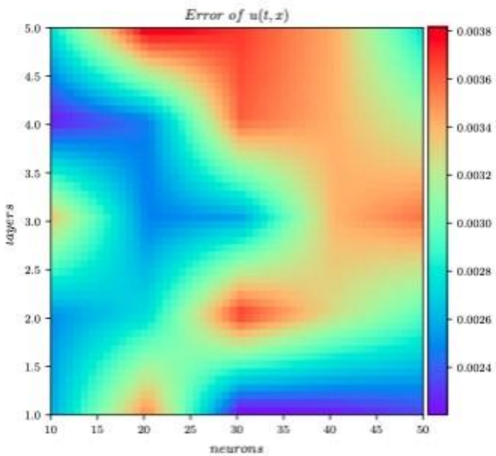
(b) L^2-error vs. neurons per layer (the number of hidden layers is fixed at 5)

Performances of different architectural designs obtained by varying the number of hidden layers and the number of neurons per layer.

Error visualization in NN hyper-parameter space

NN model error

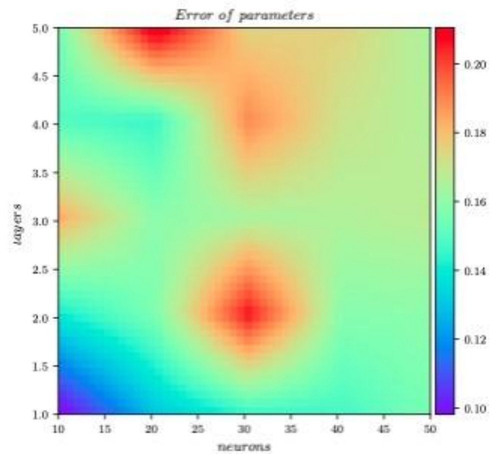
the number of layers



the number of neurons

PDE derivation error

the number of layers



the number of neurons

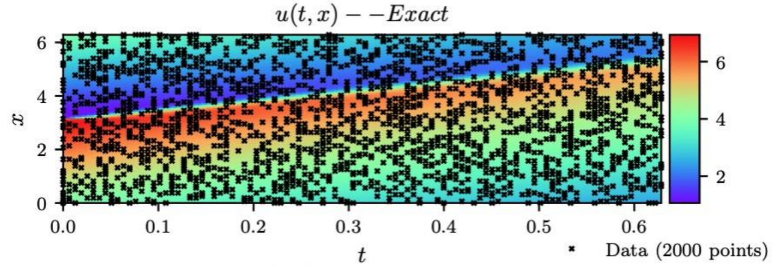
Case study: Advection-diffusion(Ad)equation

The training data set is generated from a specific exact solution that describes one initial condition.

$$\phi(0, x) = 0 \quad (x \geq 0)$$

Ad equation is a combination of the diffusion and convection (advection) equations, and describes physical phenomena where particles, energy, or other physical quantities are transferred inside a physical system due to two processes: diffusion and convection.

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = D\frac{\partial^2 \phi}{\partial x^2}$$

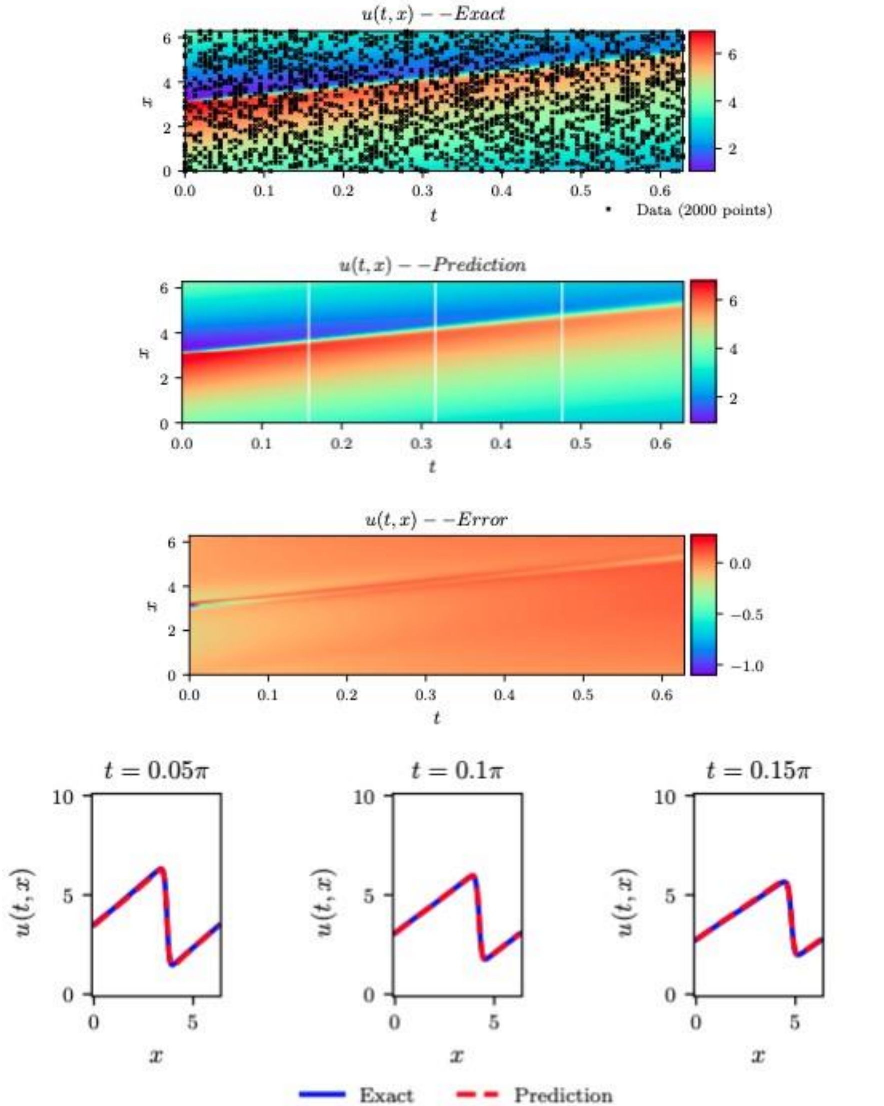


Exact solution:

$$\phi(t,x) = \frac{1}{2}\exp\left(\frac{c}{2D}x\right)\left[\exp\left(-\frac{c}{2D}x\right)\mathrm{erfc}\left(\frac{1}{2\sqrt{Dt}}(x-ct)\right) + \exp\left(\frac{c}{2D}x\right)\mathrm{erfc}\left(\frac{1}{2\sqrt{Dt}}(x+ct)\right)\right]$$

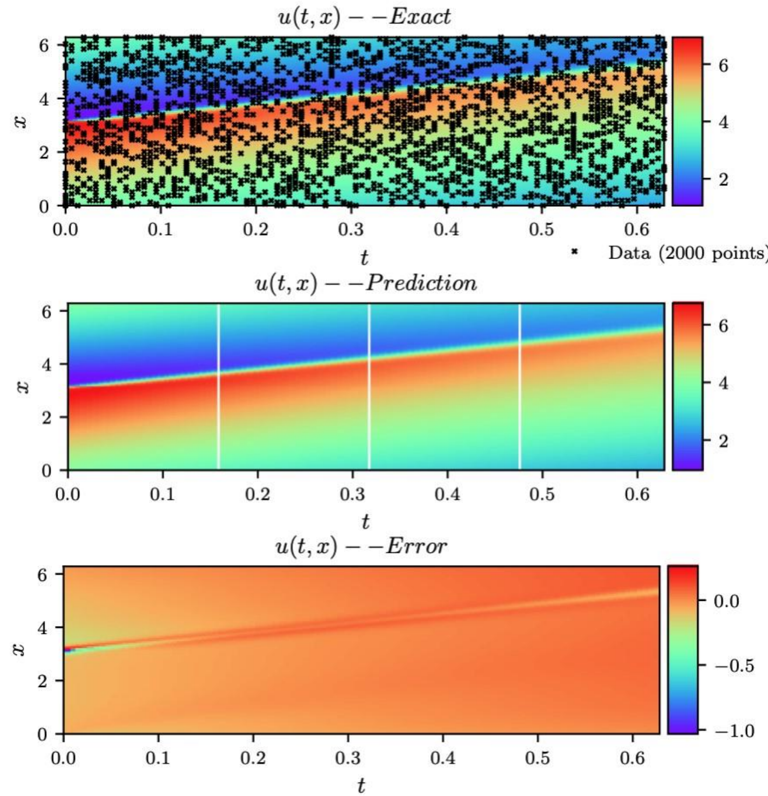
Rudy S H, Brunton S L, Proctor J L, et al. Data-driven discovery of partial differential equations[J]. Science Advances, 2017, 3(4): e1602614.

Derived results from the minimum partial derivative library

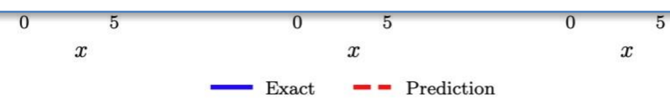


Correct PDE	$u_t + uu_x - 0.1u_{xx} = 0$
Identified PDE (clean data)	$u_t + 0.90576uu_x + -0.1597469u_{xx} = 0$

Derived result by extended partial derivative library



Is this a new PDE created by the limitation of boundary conditions?

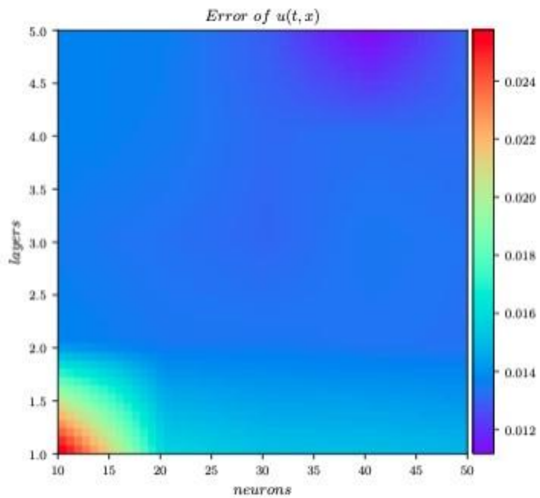


Correct PDE	$u_t + uu_x - 0.1u_{xx} = 0$
Identified PDE (clean data)	$u_t + -0.54061u_x + 1.04304uu_x + -0.1713265u_{xx} = 0$

Error visualization in NN hyper-parameter space

NN model error

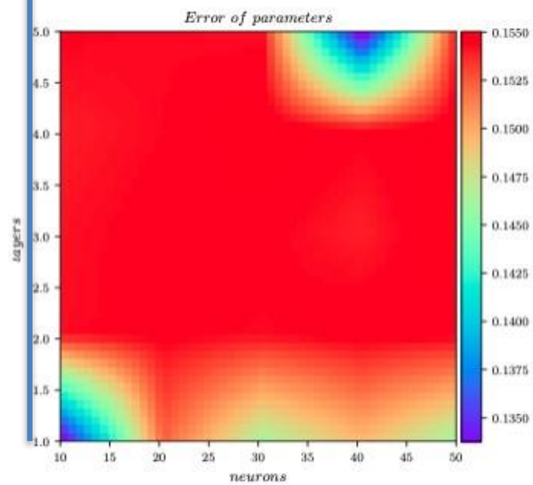
the number of layers



the number of neurons

PDE derivation error

the number of layers



the number of neurons

Case study

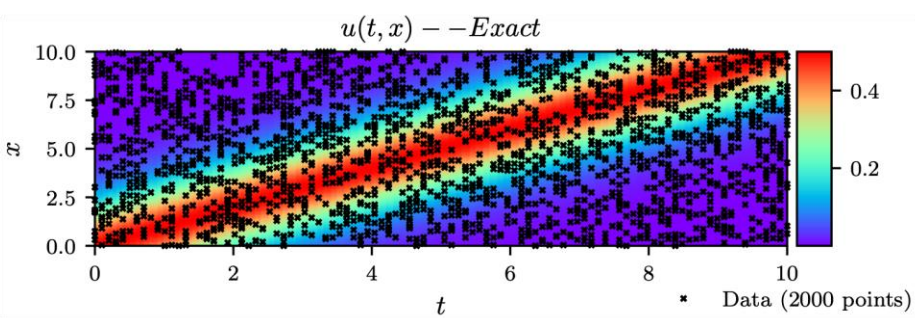
The training data set is generated from exact solutions of a PDE

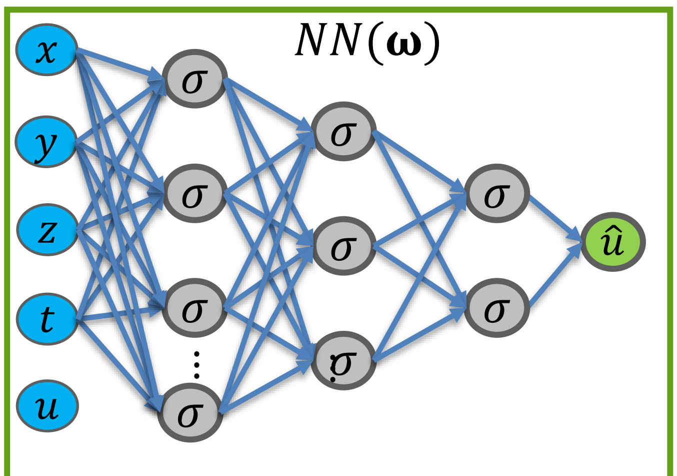
Random sampling is conducted inside a given spatio-temporal region

General neural networks are employed

The PDE derivation errors are visualized in a parameter space.

- Number of layers(N_l): $1 \leq N_l \leq 10$
- Number of neurons(N_n): $1 \leq N_n \leq 512$





Hyper-parameter space

Error distribution in a parameter space

The PDE derivation errors and NN model errors are visualized using a multi-dimensional visualization technique.

If we have the following restrictions:

- Number of layers(N_l): $1 \leq N_l \leq 10$
- Number of neurons(N_n): $1 \leq N_n \leq 512$

The errors are represented as

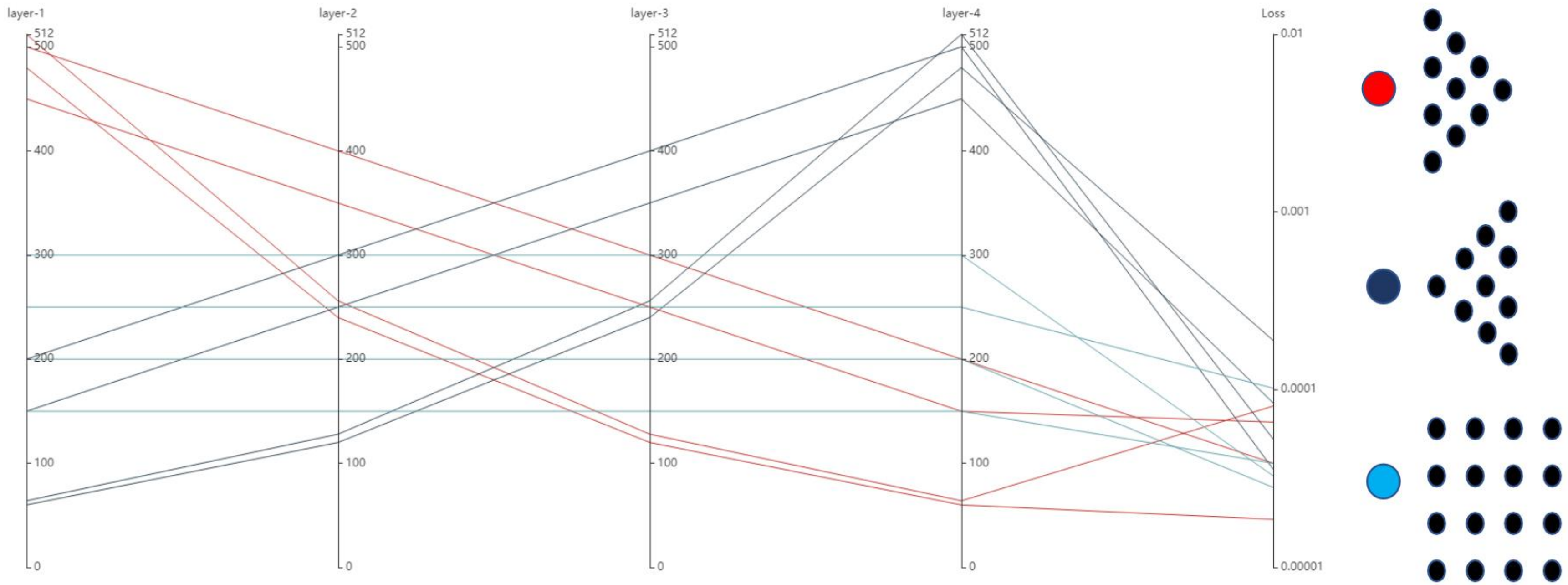
- $E_\varepsilon(N_n^1, N_n^2, N_n^3, N_n^4, N_n^5, N_n^6, N_n^7, N_n^8, N_n^9, N_n^{10})$
- $MSE_u(N_n^1, N_n^2, N_n^3, N_n^4, N_n^5, N_n^6, N_n^7, N_n^8, N_n^9, N_n^{10})$

	1	2	3	4	5	6	7	8	9	10	E_ε
1	334	443	86	130	129	386	79	204	366	113	0.989678
2	147	96	136	386	227	171	0	0	0	0	0.492013
3	52	283	114	457	79	179	494	0	0	0	0.090311
4	350	100	227	424	284	341	345	505	148	346	0.886385
…	…	…	…	…	…	…	…	…	…	…	…
1234	376	89	397	250	0	0	0	0	0	0	0.889185

Error visualization using PCs

Optimization of NN structure

If the accuracy of the NN model is improved by changing the NN hyper-parameters, it is possible to clarify the requirements of the NN structure that can maximize the accuracy of both the derivation of the partial differential equation and the NN model.



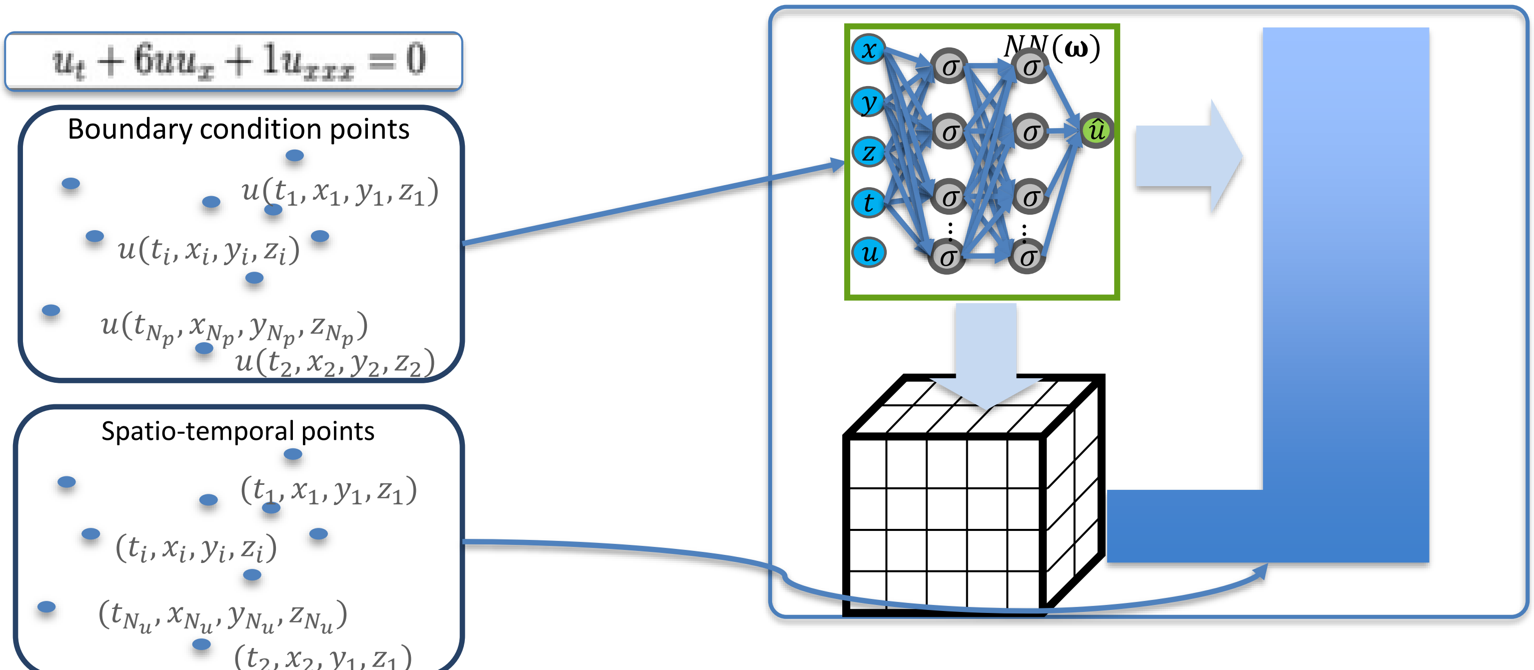
Visual data science research

PDE SOLUTION FROM SPATIO-TEMPORAL DATA

PDE solution

Background

- Previously, a local interpolation has been employed to evaluate the partial differential terms
- Effect from outside the local region cannot be considered.
- A global approximation has been expected for the evaluation.



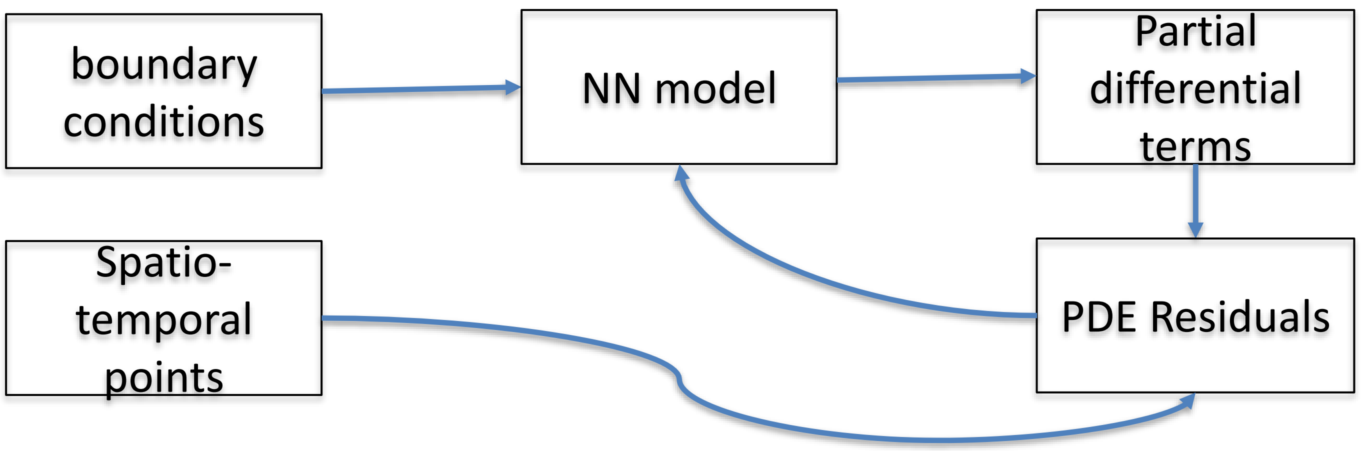
Research question: Can we solve a PDE from given initial and boundary conditions?

Proposed method: If we add initial and boundary condition error terms to the loss function, the question can be answered.

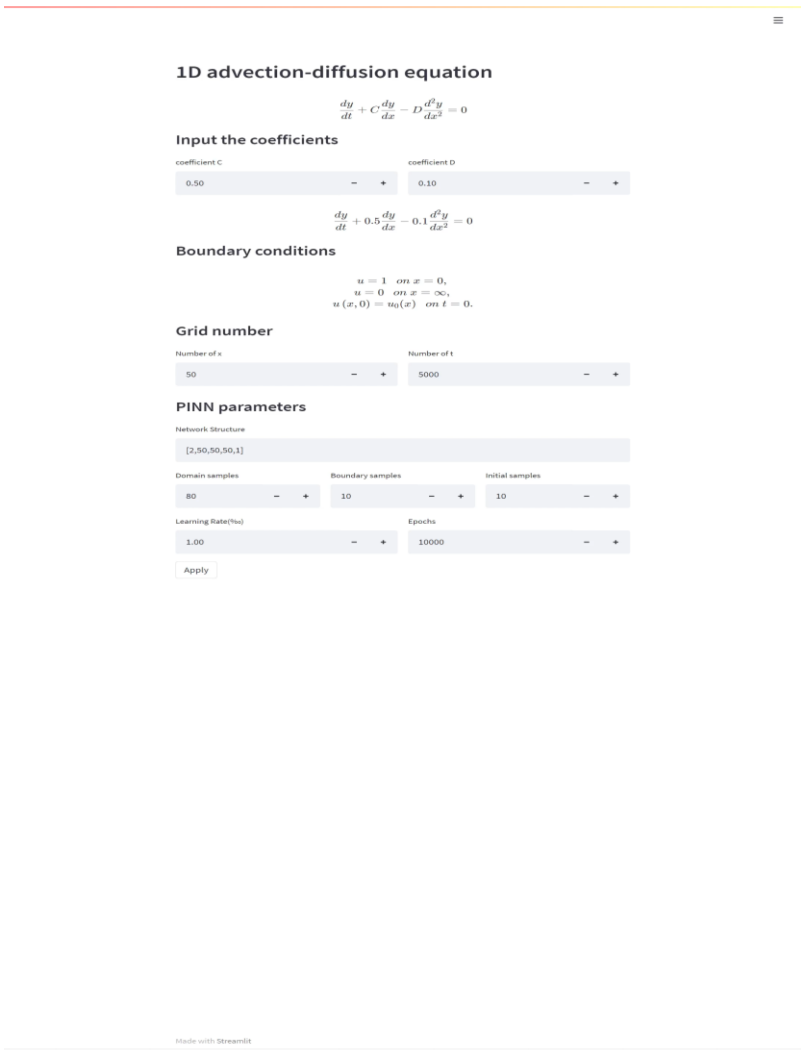
Proposed method

Solution of partial differential equations(PDEs) based on a given boundary conditions

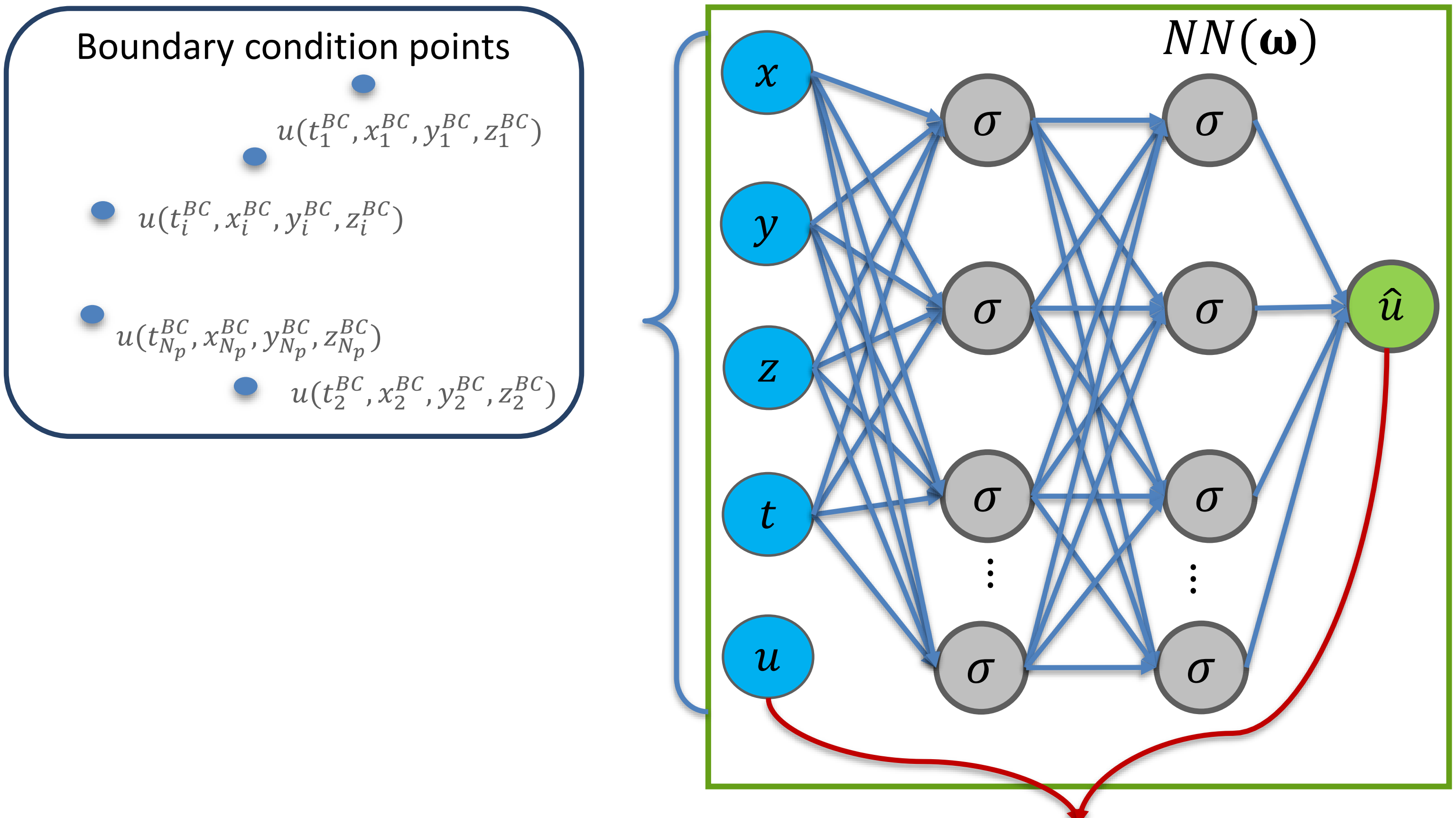
- Partial differential terms are given



Overview


1D advection-diffusion equation
$\frac{dy}{dt} + C\frac{dy}{dx} - D\frac{d^2y}{dx^2} = 0$
Input the coefficients
coefficient C
0.50
coefficient D
0.10
$\frac{dy}{dt} + 0.5\frac{dy}{dx} - 0.1\frac{d^2y}{dx^2} = 0$
Boundary conditions
$u = 1 \quad on\ x = 0,$
$u = 0 \quad on\ x = \infty,$
$u\left(x, 0\right) = u_0(x) \quad on\ t = 0.$
Grid number
Number of x
50
Number of t
5000
PINN parameters
Network Structure
[2,50,50,50,1]
Domain samples
80
Boundary samples
10
Initial samples
10
Learning Rate(‰)
1.00
Epochs
10000
Apply
Made with Streamlit

NN model from Spatio-temporal datasets

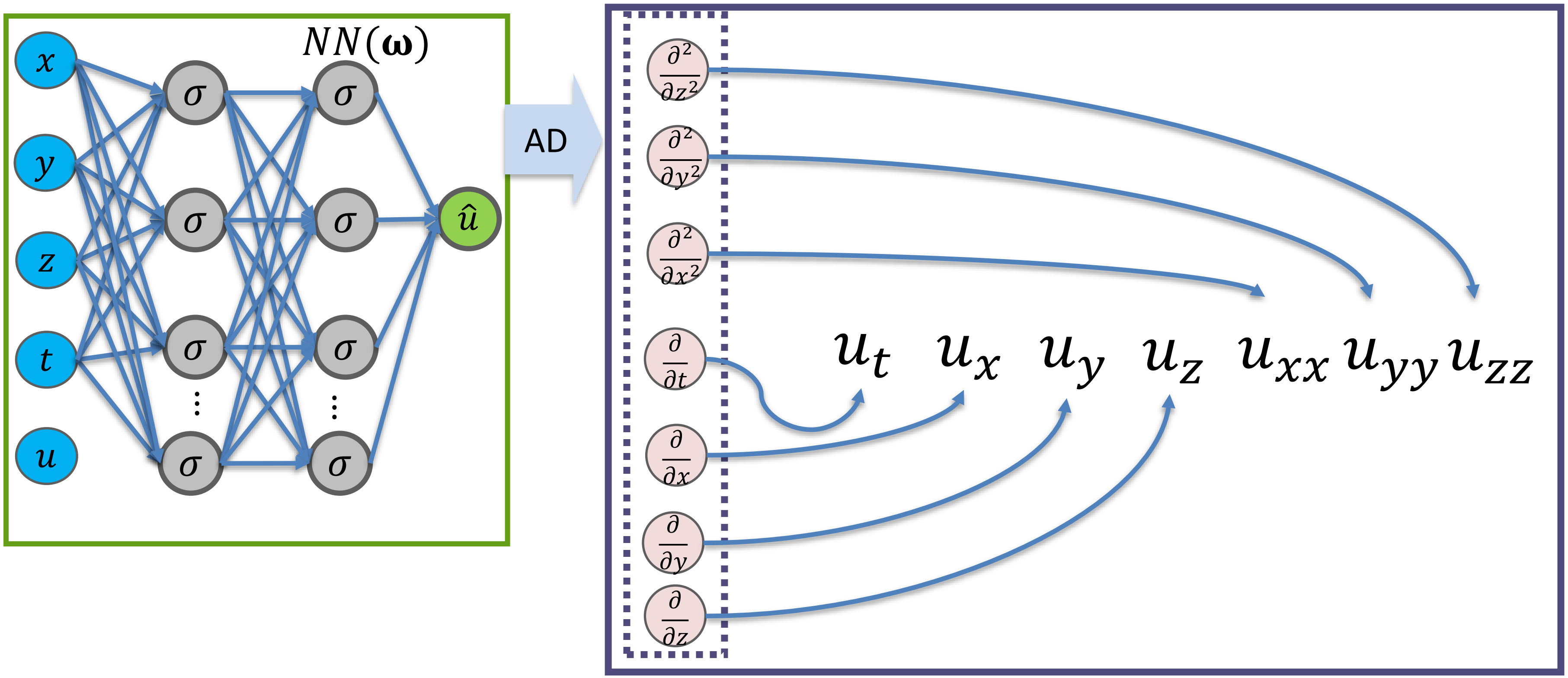


$$MSE_u = \frac{1}{N_u} ||u - \hat{u}||_2^2$$

Boundary condition points

NN model

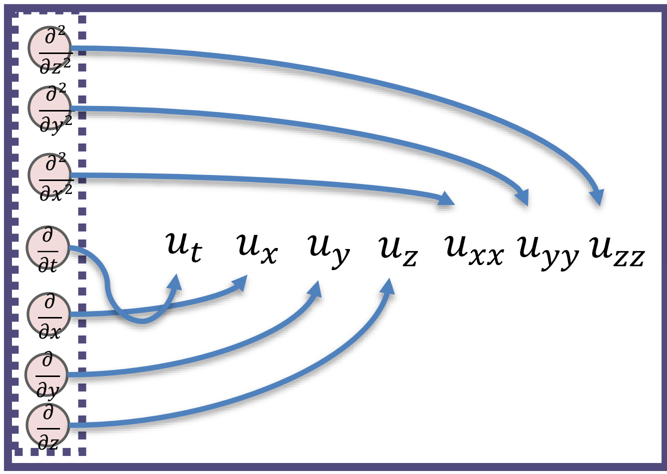
Partial differential terms from NN model



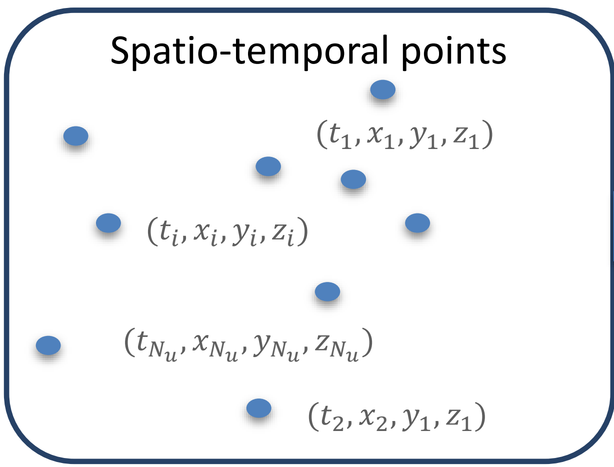
NN model

Partial differential terms

Regression equation from partial differential terms



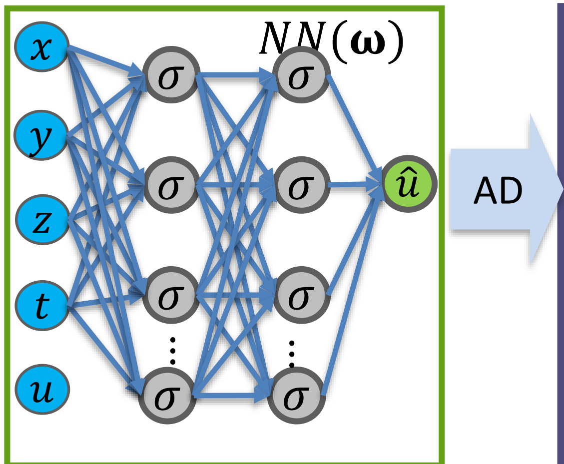
$$
\underbrace{\begin{pmatrix} u_t^1 \\ u_t^2 \\ u_t^3 \\ \vdots \\ u_t^{N_u-2} \\ u_t^{N_u-1} \\ u_t^{N_u} \end{pmatrix}}_{\mathbf{u}_t}
=
\underbrace{\begin{pmatrix}
1 & u_x^1 & u_y^1 & u_z^1 & u_{xx}^1 & u_{yy}^1 & u_{zz}^1 \\
1 & u_x^2 & u_y^2 & u_z^2 & u_{xx}^2 & u_{yy}^2 & u_{zz}^2 \\
1 & u_x^3 & u_y^3 & u_z^3 & u_{xx}^3 & u_{yy}^3 & u_{zz}^3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & u_x^{N_u-2} & u_y^{N_u-2} & u_z^{N_u-2} & u_{xx}^{N_u-2} & u_{yy}^{N_u-2} & u_{zz}^{N_u-2} \\
1 & u_x^{N_u-1} & u_y^{N_u-1} & u_z^{N_u-1} & u_{xx}^{N_u-1} & u_{yy}^{N_u-1} & u_{zz}^{N_u-1} \\
1 & u_x^{N_u} & u_y^{N_u} & u_z^{N_u} & u_{xx}^{N_u} & u_{yy}^{N_u} & u_{zz}^{N_u}
\end{pmatrix}}_{\boldsymbol{\Theta}}
\underbrace{\begin{pmatrix} \varepsilon \\ \varepsilon_{u_x} \\ \varepsilon_{u_y} \\ \varepsilon_{u_z} \\ \varepsilon_{u_{xx}} \\ \varepsilon_{u_{yy}} \\ \varepsilon_{u_{zz}} \end{pmatrix}}_{\boldsymbol{\varepsilon}}
$$

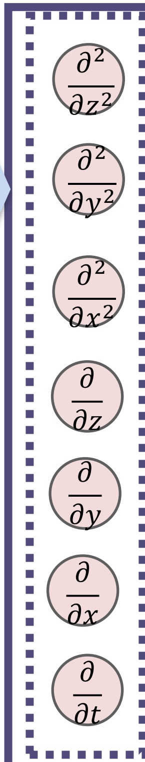


Partial differential terms

Regression equation

PDE residuals





$$\mathbf{u}_t - \boldsymbol{\Theta}\boldsymbol{\varepsilon} = \mathbf{0}$$

$$MSE_{PDE} = \frac{1}{N_u}||\mathbf{u_t} - \boldsymbol{\Theta}\boldsymbol{\varepsilon}||_2^2$$

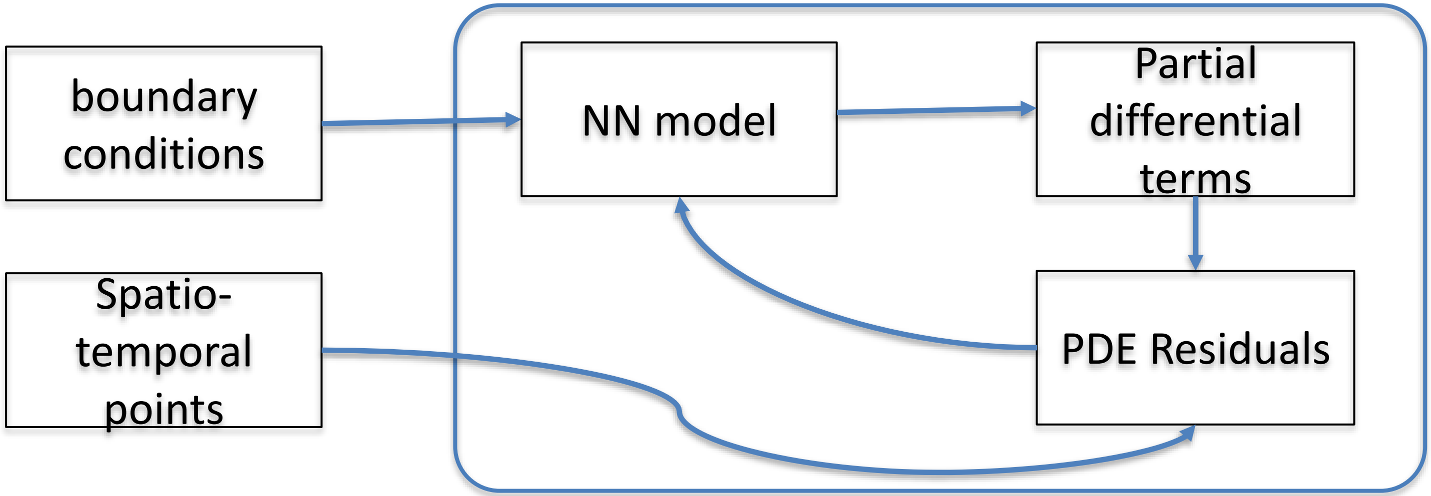
NN model

PDE residuals

Physics informed NN(PINN)

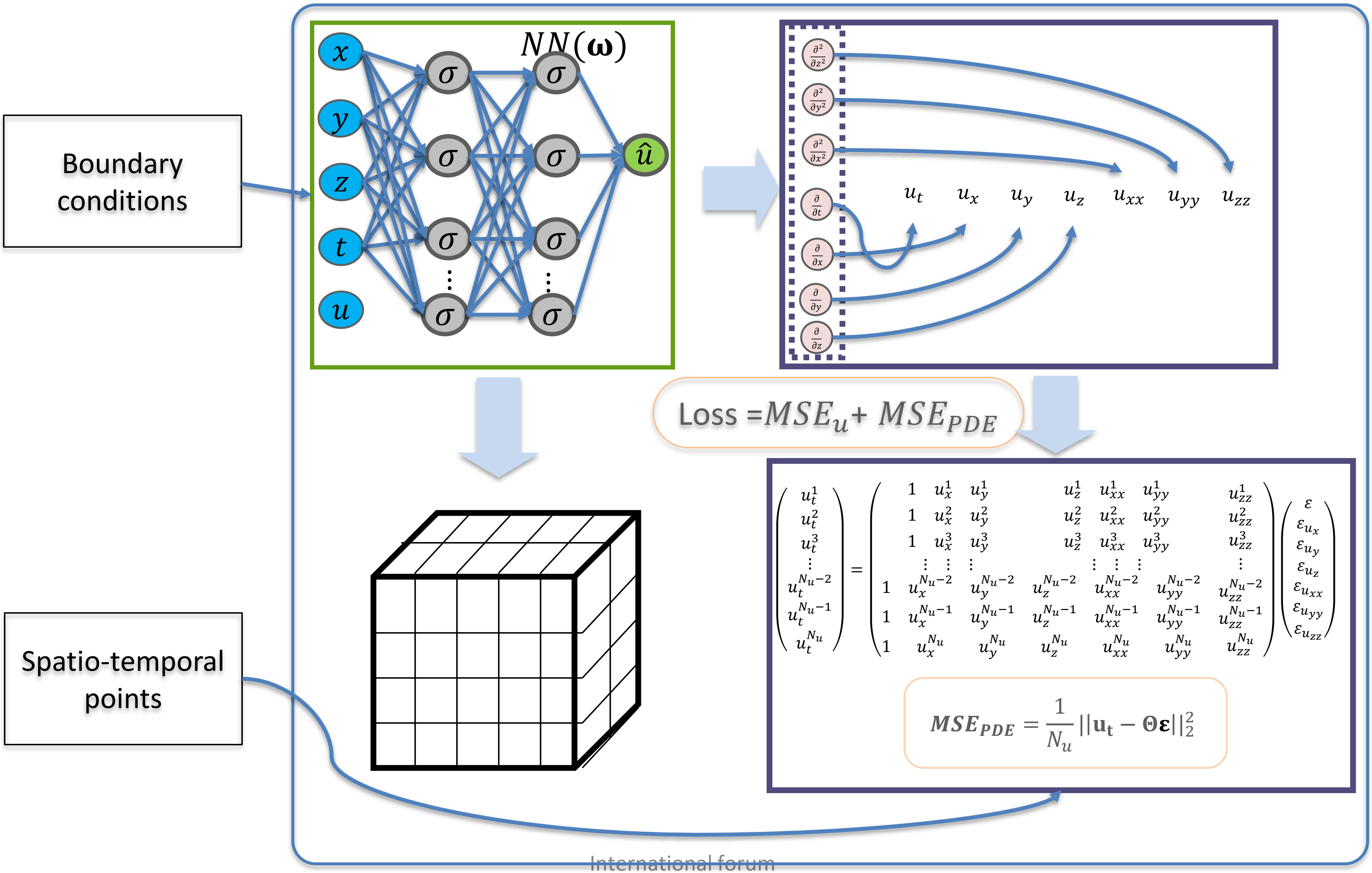
Add MSE_{PDE} to the loss function of NN model

- Adequate training of NN model
- Multi-objective optimization



M. Raissi et. al,"Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." Journal of Computational Physic, 378, 686-707, 2019

PDE solution



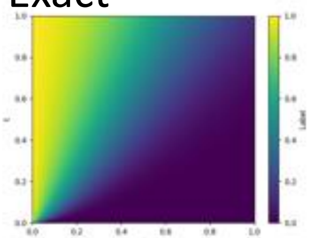
Case study: Ad equation

The training data set is generated from a specific exact solution that describes one initial condition.

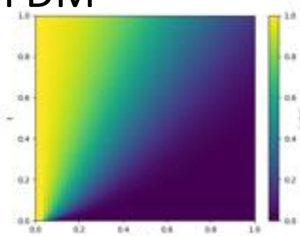
$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = D\frac{\partial^2 \phi}{\partial x^2} \qquad \phi(0,x) = 0 \quad (x \geq 0)$$

The equation is a combination of the diffusion and convection (advection) equations, and describes physical phenomena where particles, energy, or other physical quantities are transferred inside a physical system due to two processes: diffusion and convection. .

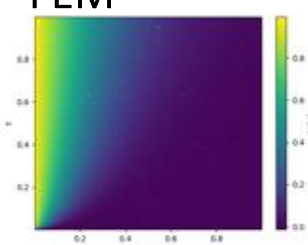
Exact



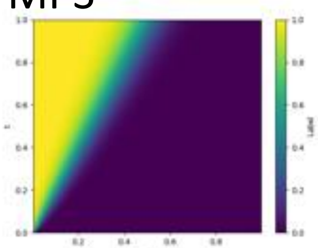
FDM



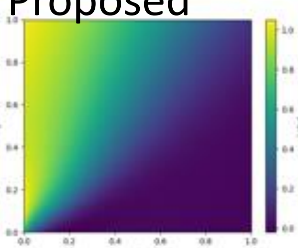
FEM



MPS



Proposed



FDM	2.47×10^{-2}	**FEM**	1.54×10^{-1}
MPS	1.27×10^{-1}	**Proposed**	1.42×10^{-2}

Exact solution:

$$\phi(t,x) = \frac{1}{2}\exp\left(\frac{c}{2D}x\right)\left[\exp\left(-\frac{c}{2D}x\right)\mathrm{erfc}\left(\frac{1}{2\sqrt{Dt}}(x-ct)\right) + \exp\left(\frac{c}{2D}x\right)\mathrm{erfc}\left(\frac{1}{2\sqrt{Dt}}(x+ct)\right)\right]$$

PDE solution system

Equation properties

- Equation form / cofficients
- Boundary (/ initial) conditions

PINN properties

- Network structure
- Network structure
- Number of samples
  - Domain
  - Boundary
  - Initial
- Learning rate
- Epochs

Resolution of solution image

Visualization

Solution images

(Comparison)
Error images

Loss images

Visualization System (Streamlit)

1D advection-diffusion equation

$$\frac{dy}{dt} + C\frac{dy}{dx} - D\frac{d^2y}{dx^2} = 0$$

Input the coefficients

coefficient C

0.50 − +

coefficient D

0.10 − +

$$\frac{dy}{dt} + 0.5\frac{dy}{dx} - 0.1\frac{d^2y}{dx^2} = 0$$

Boundary conditions

$$\begin{aligned} u = 1 \quad on\ x = 0, \\ u = 0 \quad on\ x = \infty, \\ u\left(x, 0\right) = u_0(x) \quad on\ t = 0. \end{aligned}$$

Grid number

Number of x

50 − +

Number of t

5000 − +

PINN parameters

Network Structure

[2,50,50,50,1]

Domain samples

80 − +

Boundary samples

10 − +

Initial samples

10 − +

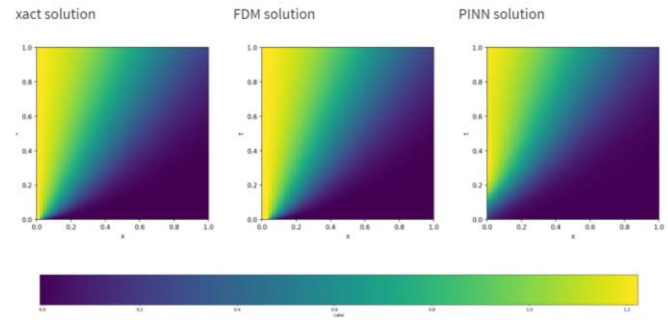
Learning Rate(‰)

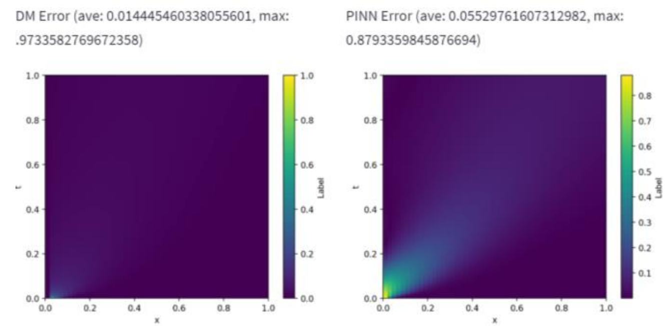
1.00 − +

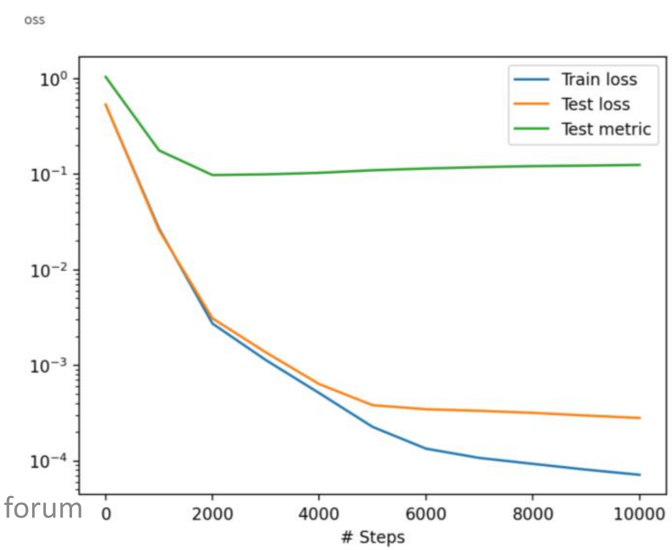
Epochs

10000 − +

Apply

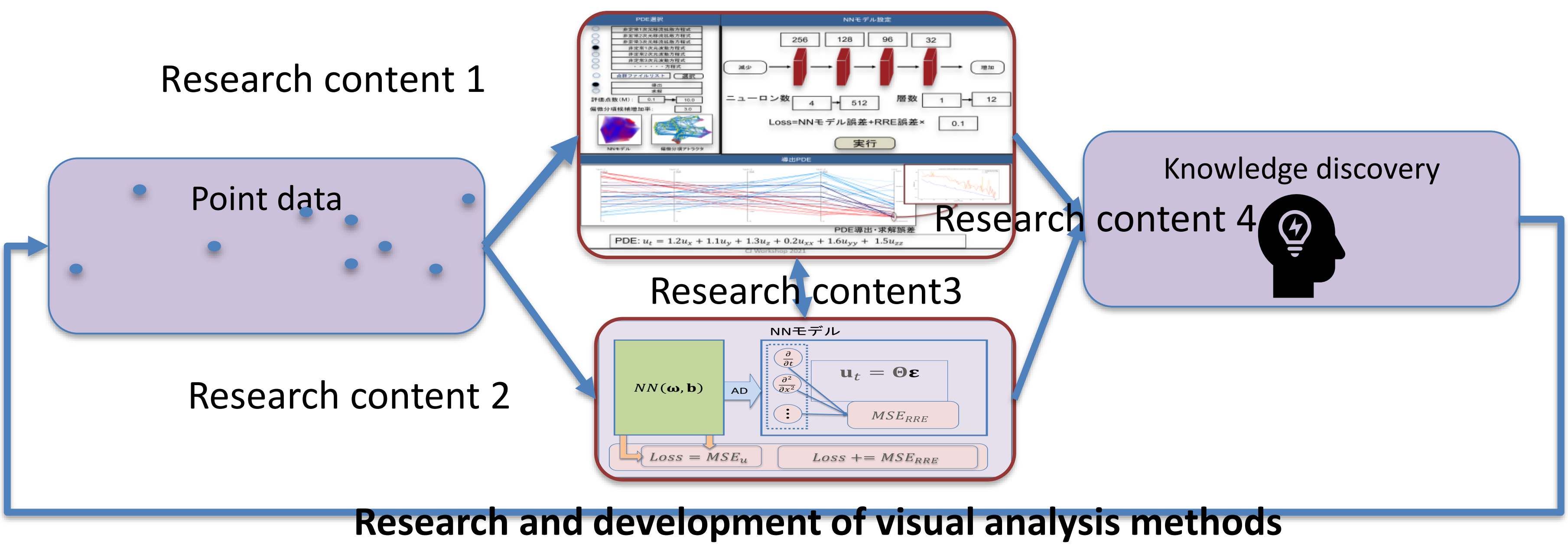




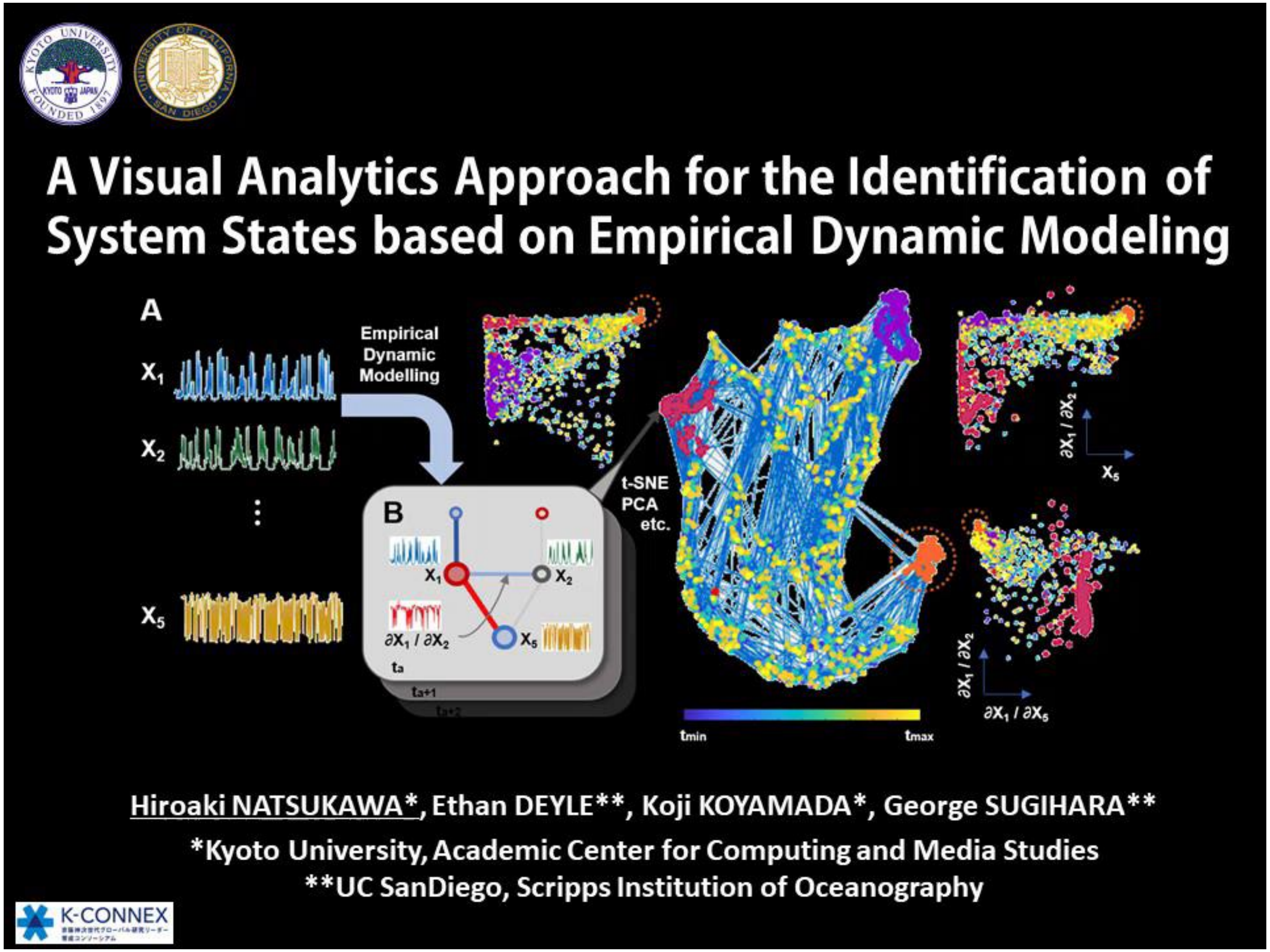


Visual analysis and its requirements

1. Clarify how much the point data and the constructed NN model match in spatio-temporal space (Research contents 1, 2)
2. Clarify how the diversity in the partial differential terms and boundary conditions decrease the PDE derivation error (Research Content 3)
3. Clarify how the NN model parameters and the number of points decrease the PDE derivation and solution error (Research content 3)
4. Analyze the PDE solution error in local and global regions, with conventional methods (Research content 4)
5. Clarify how to select the partially differentiated term candidates using EDM.



Empirical dynamic modeling(EDM)



Summary

Our visual data science meets AI through PINN

- Universal function approximation
- AD for partial differentiation terms

Application examples

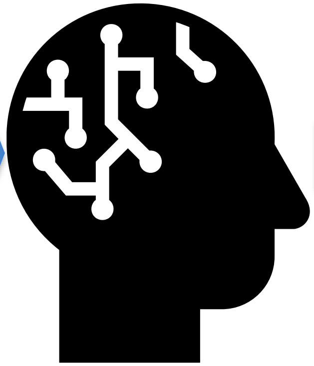
- Page extraction from booklet data captured by 3D-CT
- Extraction of plasma region from electromagnetic field analysis results in fusion reactor
- Derivation and solution of PDEs

Rendering of a neural network volume

- Indirect approach by mapping the volume to grid
- Direct approach, ray-tracing using automatic integration
- David, et al,"AutoInt: Automatic Integration for Fast Neural Volume Rendering" CVPR2021

Diversity and inclusion

1. We should consider diversity to share a future
2. Diversity requires tremendous mount of different conditions



https://www.generationsforpeace.org/en/op-ed-how-can-cultural-diversity-drive-peace-and-development/

https://www.kensetsunews.com/web-kan/502950