2025 Korea Nuclear Society

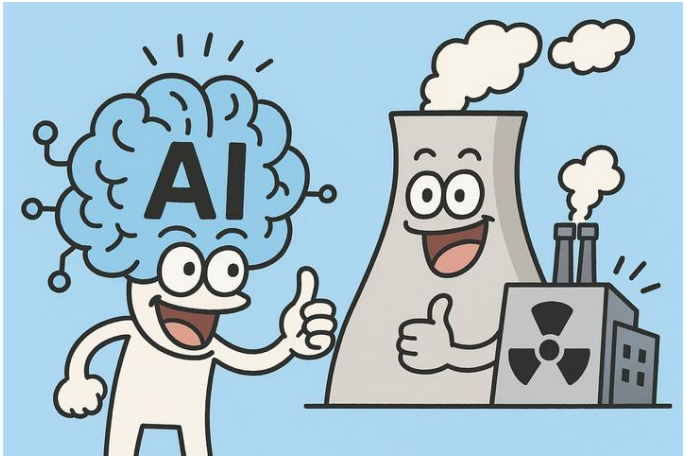
Preliminary CFD Benchmarking of L-DeepONet for Nuclear Reactor Digital Twin

Minseo Lee [a], Sangam Karnal [a], Shilaj Baral [a], Minseop Song [b], Chaehyeon Song [b], Joongoo Jeon [a]*

NINE Lab,

[a] Department of Applied Plasma and Quantum Beam Engineering, Jeonbuk National University,

[b] Department of Nuclear Engineering, Hanyang University



2025.05.22.

Contents

Contents

Introduction

Computational Fluid Dynamics

» Computational Fluid Dynamics (CFD) is a method to simulate the flow of liquids and gases based on the conservation equations of mass, momentum, and energy using computers.

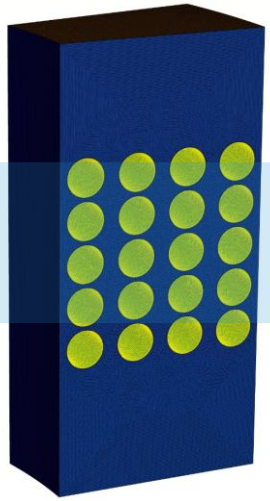
① Mesh generation

② Select Governing Equations

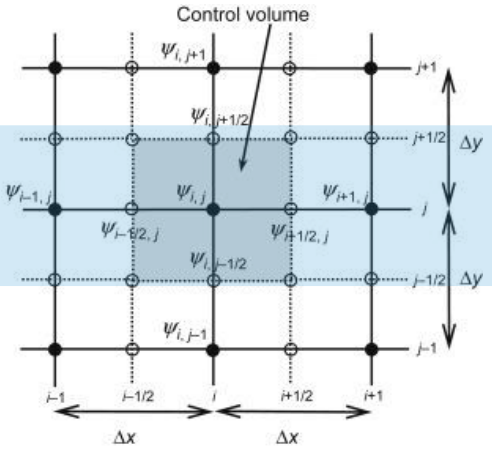
③ Using the numerical methods like FVM, FEM, FDM to solve PDEs

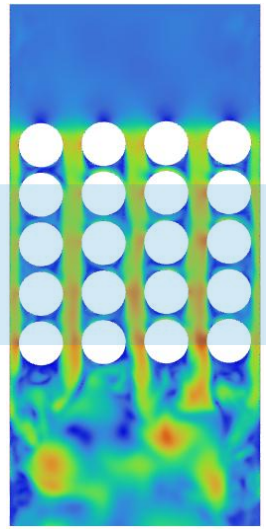
④ Simulation

Too expensive!!



$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f}_b$$



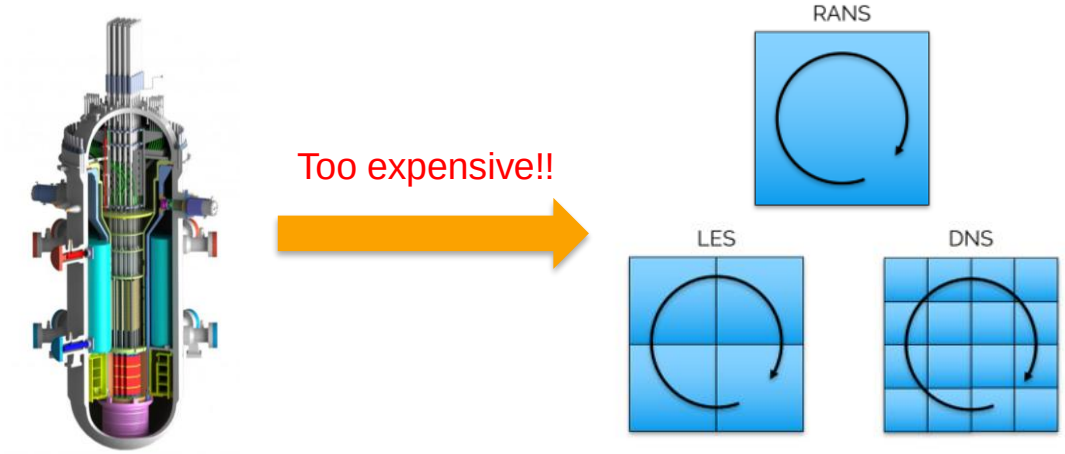


Introduction

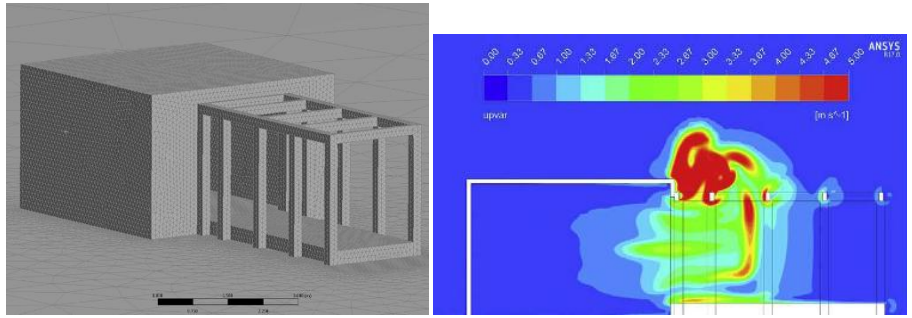
Disadvantages of CFD

- Fluid phenomena within nuclear reactors involve turbulence flow and heat transfer occurring simultaneously in complex geometries.
- Calculating at DNS and LES levels requires a dense grid, but such grid scales are realistically impossible at the reactor scale.
- In the case of reactive flows such as hydrogen combustion, the calculation time required is very long due to the discrepancy between the chemical reaction time within the fluid and the flow time scales.

❖ Turbulence flow



❖ hydrogen combustion→ 1s simulation : 100 hours(Tolias et al., 2018)



Code	Computational Domain	Simulation Time
FLACS	20 × 14.4 × 12 m (Rectangular box)	6 h for 1 s
CFX	100 m (Cubic box)	48 h for 0.6 s
FLUENT	Hemispherical radius 25 m	80 h for 0.45 s
ADREA-HF	42 × 60 × 30 m (Rectangular box)	315 h for 0.45 s

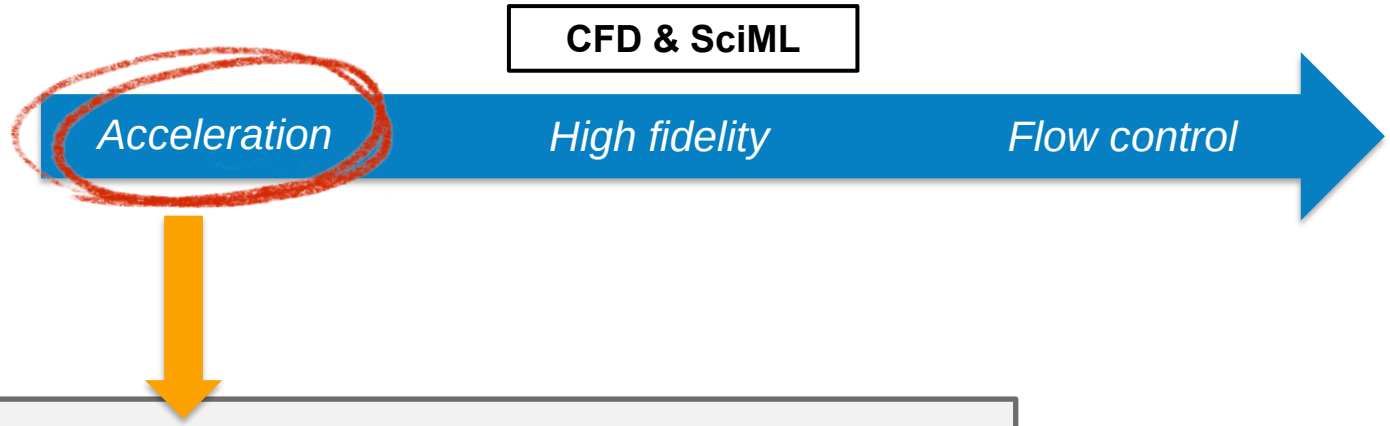
Tolias, I. C., et al. "Numerical simulations of vented hydrogen deflagration in a medium-scale enclosure." Journal of loss prevention in the process industries 52 (2018): 125-139.

Introduction

SciML?

- Scientific Machine Learning(SciML) : A method of approach and interpretation utilizing machine learning based on scientific knowledge and data.
- Why SciML in CFD?
  - ① Reduced **simulation time** compared to existing numerical methods
  - ② Increased **analysis accuracy** in turbulence models
  - ③ Furthermore, it can also be utilized to optimal flow control tasks.

- SciML Model
  - Physics Informed Neural Network
  - Operator Learning
  - ML-CFD Hybrid Solver



This research examines SciML applicability to reactor geometry and to perform CFD acceleration research for digital twins.

Contents

Method

Operator Learning

≫ Operator Learning is learning the operator G that takes function u as input and outputs function s, through an artificial neural network.

$$f_1 \xrightarrow[\text{(Operator)}]{G} f_2 \qquad f_2 = \frac{df_1(x)}{dx} = \frac{d}{dx} sin(x) = cos(x) \qquad u(\cdot) \dashrightarrow s(\cdot)$$

(Approximate)

≫ **Universal Approximation Theorem for Operator**?

: The Universal Approximation Theorem for Operators states that a feedforward neural network with a single hidden layer containing a finite number of neurons can approximate any continuous function to an arbitrary degree of accuracy, provided that the activation function is non-linear.

$$G(u)(y) - \sum_{k=1}^{P} \sum_{i=1}^{n} c_{\kappa i} \sigma\left(\sum_{i}^{m} \xi_{ij}^{k} u(x_j) + \theta_i^k\right) \sigma(w_k \cdot y + \zeta_k) < \epsilon$$

✓ Neural Operator Model

1. Deep Operator Network(DeepONet)
2. Fourier Neural Operator
3. Laplace Neural Operator

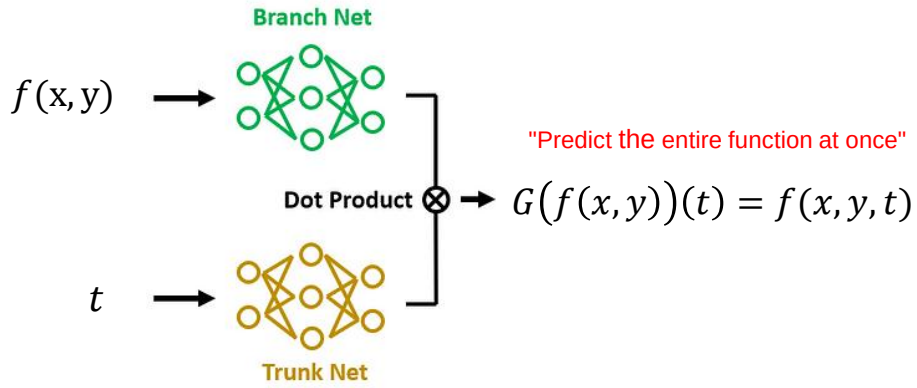
https://medium.com/data-science/operator-learning-via-physics-informed-deeponet-lets-implement-it-from-scratch-6659f3179887#5221

Method

DeepONet

- DeepONet consists of two components called **Branchnet** and **Trunknet**.
- Branchnet receives an **initial condition of function as input** and Trunknet takes **coordinate information** (time and space) as input.
- The output values of Branchnet and Trunknet calculated at the input coordinate (t) is predicted through the inner product operation of the two values.
- DeepONet is supervised learning that is learning by minimizing the loss between predict output and ground truth.

❖ Architecture of DeepONet



❖ Finiet Element Method(FEM) vs DeepONet

FEM Equation

$$u(x) = \sum_{i=1}^{n} c_i \phi_i(x)$$

DeepONet Equation

$$G(u)(y) \approx \sum_{k=1}^{p} b_k t_k$$

- Approximation functions as a linear combination of coefficients and basis functions
- Used fixed basis functions
- **Re-simulations** needed when input condition changes

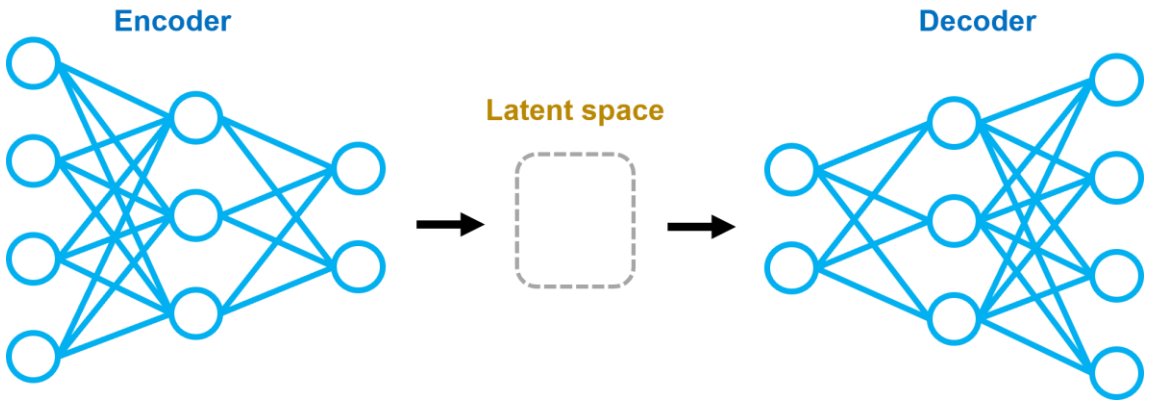
- Neural network learns coefficients and basis functions to approximate functions.
- Neural network learns optimal basis functions.
- **No re-simulations** even with changes in input conditions.

Method

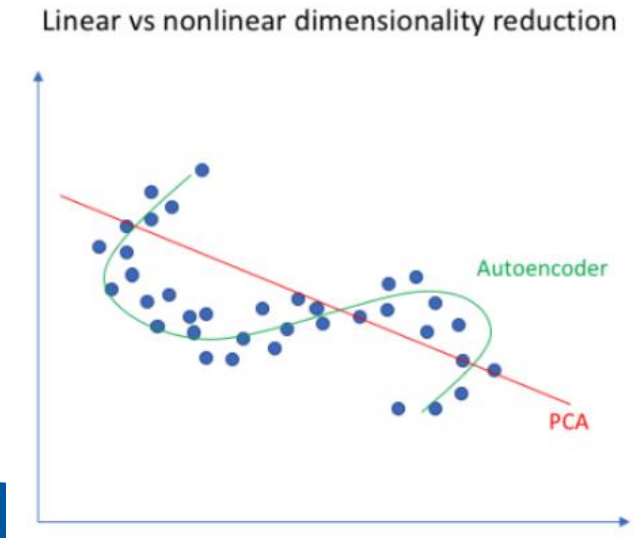
AutoEncoer

- AutoEncoder is a dimensionality reduction model that learns to map important features of data to a lower-dimensional representation through a process of compressing and restoring input data.
- Autoencoders also exhibit a tendency to prioritize learning low-frequency structures over high-frequency details, which makes them particularly effective in capturing the global characteristics of the flow
- Through the Encoder, we can obtain a low-dimensional representation that captures the overall structure of the input data.

❖ Architecture of AutoEncoder



❖ AutoEncoder vs PCA

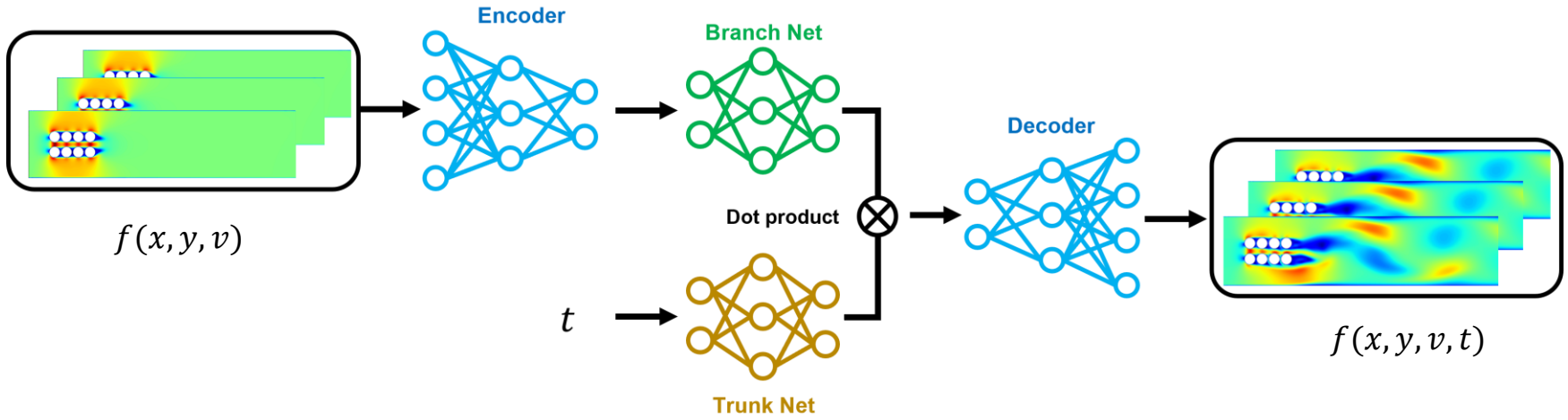


Method

Latent DeepONet

- **AutoEncoder**(Dimensionality Reduction Model) + **DeepONet** (Neural Operator)
- L-DeepONet is a model that extracts only the core information of the input function through AutoEncoders and then uses this to train and predict with DeepONet.
- By utilizing L-DeepONet, it is possible to **learn the overall characteristics** of the data while also **achieving computational efficiency**.
- Because autoencoders focus on low-frequency structures, L-DeepONet is effective for predicting global flow behavior quickly, which is a benefit in CFD acceleration modeling for complex flows.

❖Architecture of L-DeepONet



Method

Data Geometry

≫ To generate CFD flow fields, the primary side cross-section of a helical coil steam generator (HCSG) was modeled based on the **SMART** reactor design developed by the Korea Atomic Energy Research Institute (KAERI).

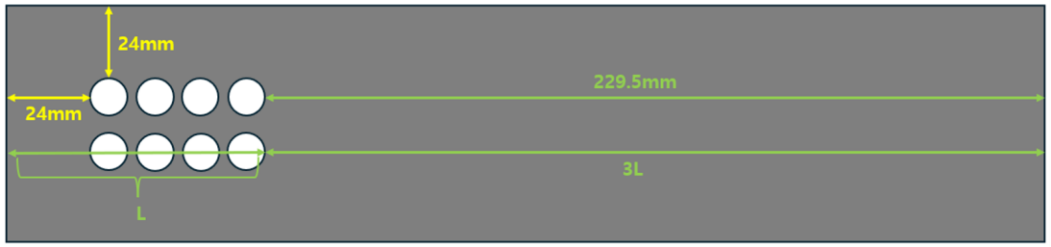
❖ Geometry 1

→ Geometry 1 is a latticed-array configuration where cylinders are aligned in straight rows to generate periodic Kármán vortex streets.

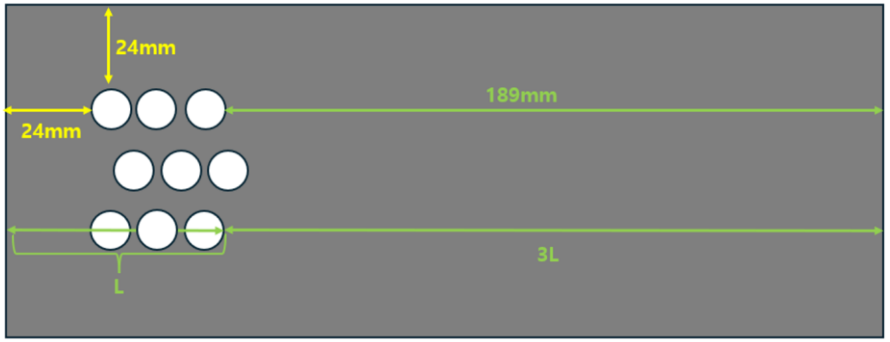
❖ Geometry2

→ Geometry 2 is a staggered-array configuration with offset rows, designed to analyze vortex interactions and wake deflection phenomena.

Geometry 1



Geometry 2



Method

Why **Kármán vortex?**

- According to the NuScale report, a comprehensive vibration assessment program was conducted to verify the structural integrity of systems and components against flow-induced vibration (FIV) mechanisms.
- Vortex shedding(VS) is one of the six flow-induced vibration mechanisms considered in the evaluation.
- Therefore, this study aims to evaluate the performance of the CFD surrogate model by analyzing the Kármán vortex, a key flow phenomenon occurring in the helical coil steam generator.

❖ Table 1. NuScale Power Module components screened for susceptibility to flow induced vibration mechanisms

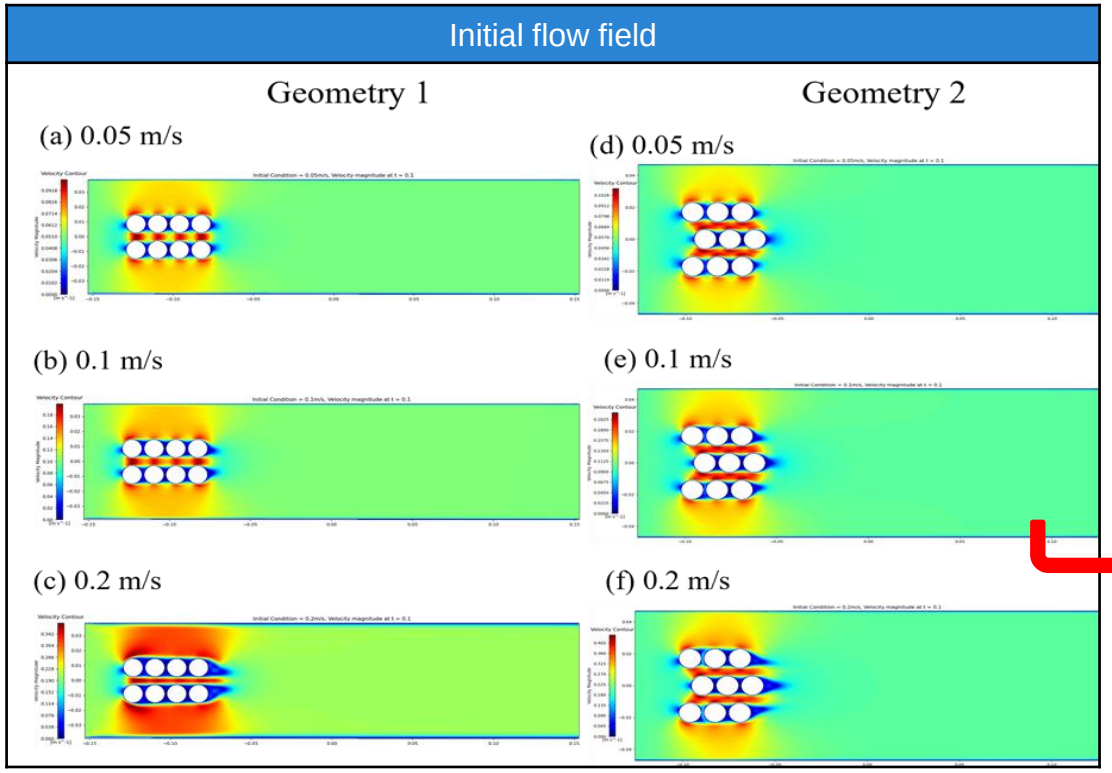
Component category	Component	Mechanisms					
		FEI	VS	TB	AR	LFI	F/G
Components exposed to secondary side flow	SG steam plenum				O		
	Helical SG tube	O	O	O			
	SG tube inlet flow restrictors			O		O	
SG tube supports	SG tube support bars			O			
	SG lower support bars		O	O			O

NuScale Power LLC. (2016), NuScale Comprehensive Vibration Assessment Program Technical Report, TR-0716-50439-NP.
[NSTAR-24NS11-512] 나선형 증기발생기 관련 규제사례 분석 및 현행 지침 적용가능성 검토

Method

CFD data generation

- CFD data were generated by applying different initial inlet velocities for each of the two geometries using Ansys Fluent.
- Simulation time : 30 minutes for each initial condition
- The flow field is defined as a function of velocity magnitude, and the discretized velocity values at each node are used as inputs.



❖ **Table 2. Training and Test Data condition**

Training Data	Test Data	
	Interpolation	Extrapolation
0.05m/s, 0.1m/s, 0.2m/s	0.15m/s	0.3m/s

Timestep	15525	15526	15527	15528
1	3.00E-01	3.00E-01	3.00E-01	3.00E-01
2	3.02E-01	3.02E-01	3.02E-01	3.02E-01
3	3.03E-01	3.03E-01	3.03E-01	3.03E-01
4	3.04E-01	3.04E-01	3.04E-01	3.04E-01
5	3.05E-01	3.05E-01	3.05E-01	3.05E-01
6	3.06E-01	3.06E-01	3.06E-01	3.06E-01
7	3.07E-01	3.07E-01	3.07E-01	3.07E-01
8	3.07E-01	3.07E-01	3.07E-01	3.07E-01

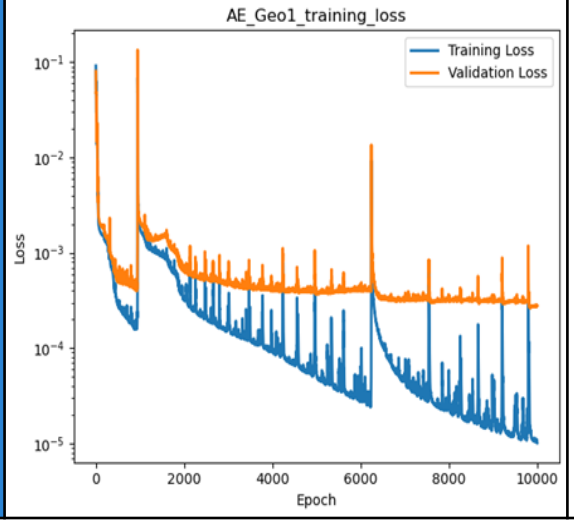
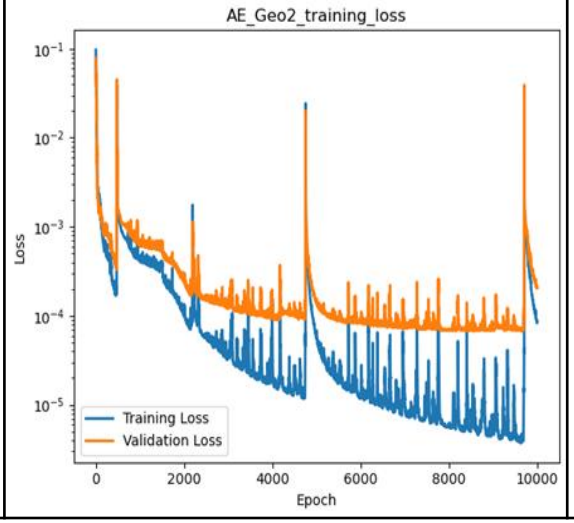
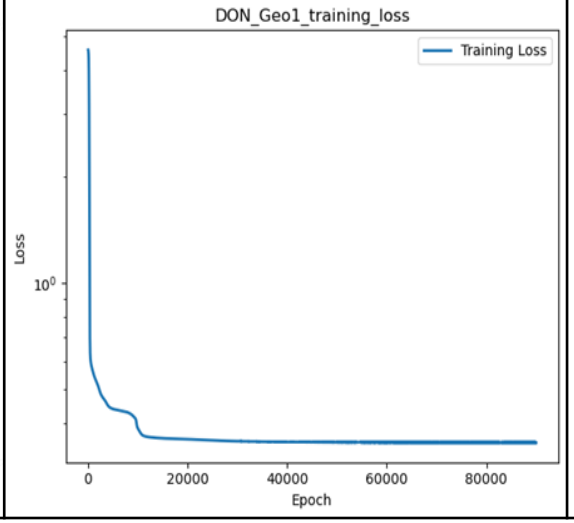
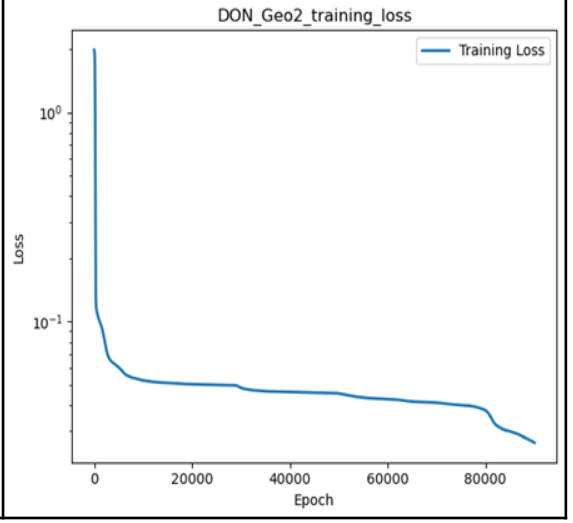
❖ **Table 3. Solver setting**

Solver	Pressure-based
Fluid Material	Water
Turbulence model	SST k-w
Tube wall condition	No-slip condition
Heat	No consider
Density	998.2 (constant)
Viscosity	0.001 (constant)
Number of Time Step	1000
Time scale factor	0.01sec

Contents

Results

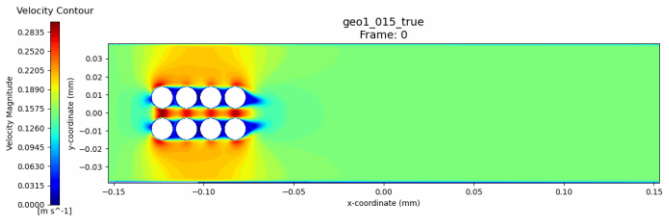
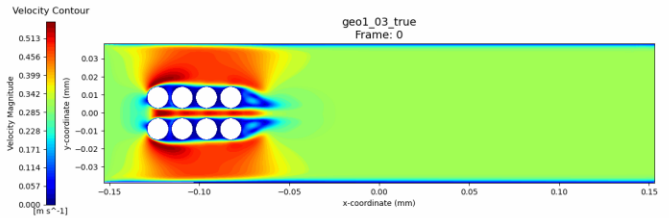
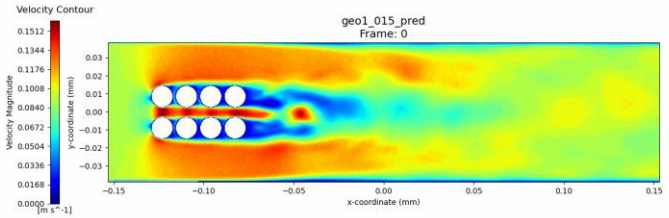
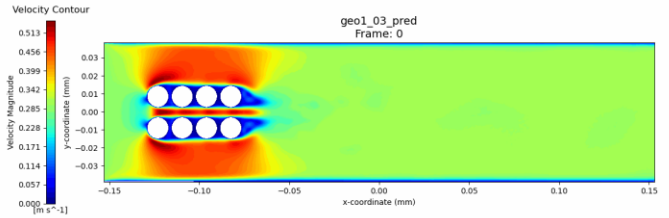
Preliminary evaluation for the application of L-DeepONet

	AutoEncoder – Geometry1	AutoEncoder – Geometry2	DeepONet – Geometry1	DeepONet – Geometry2
Loss				

Loss graph

- AutoEncoder shows similar training loss for both geometries.
- In the case of DeepONet, the training loss decreases well for both geometries, and it records even lower mean squared error (MSE) loss, especially in Geometry 2.

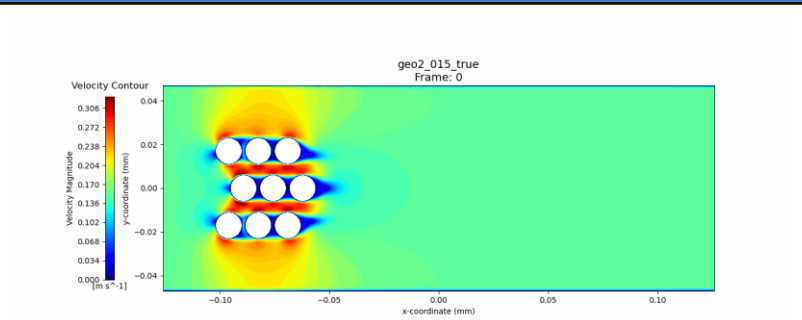
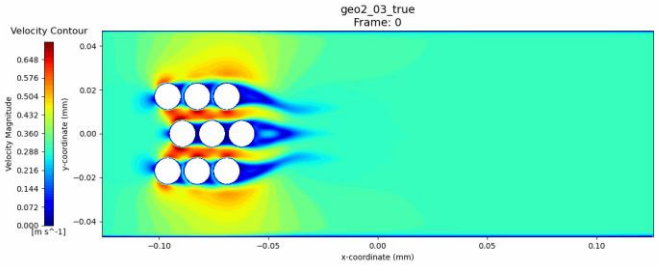
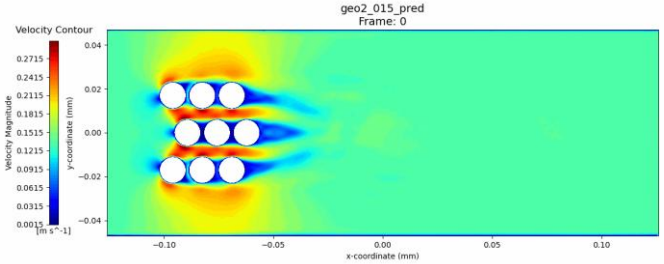
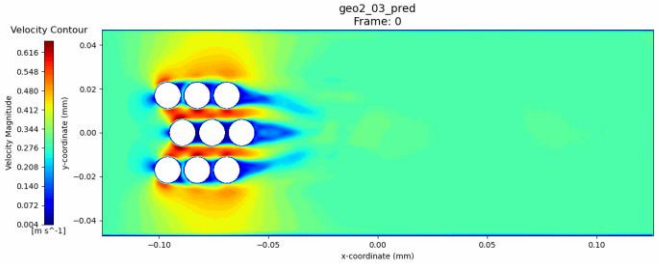
Results

	Geometry1 – 0.15m/s	Geometry1 – 0.3m/s
Ground truth		
Prediction		

Geometry1 Results

- In the case of the evaluation condition at 0.15 m/s, the model fails to accurately predict the flow field.
- In 0.3 m/s, the predicted flow initially shows similar patterns, but discrepancies become apparent as time progresses.
- Overall, the model appears to have struggled in learning the flow characteristics for the Geometry 1 configuration.

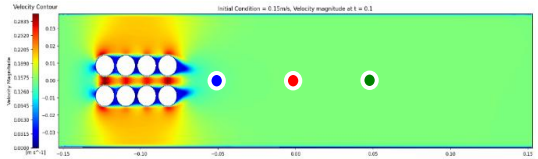
Results

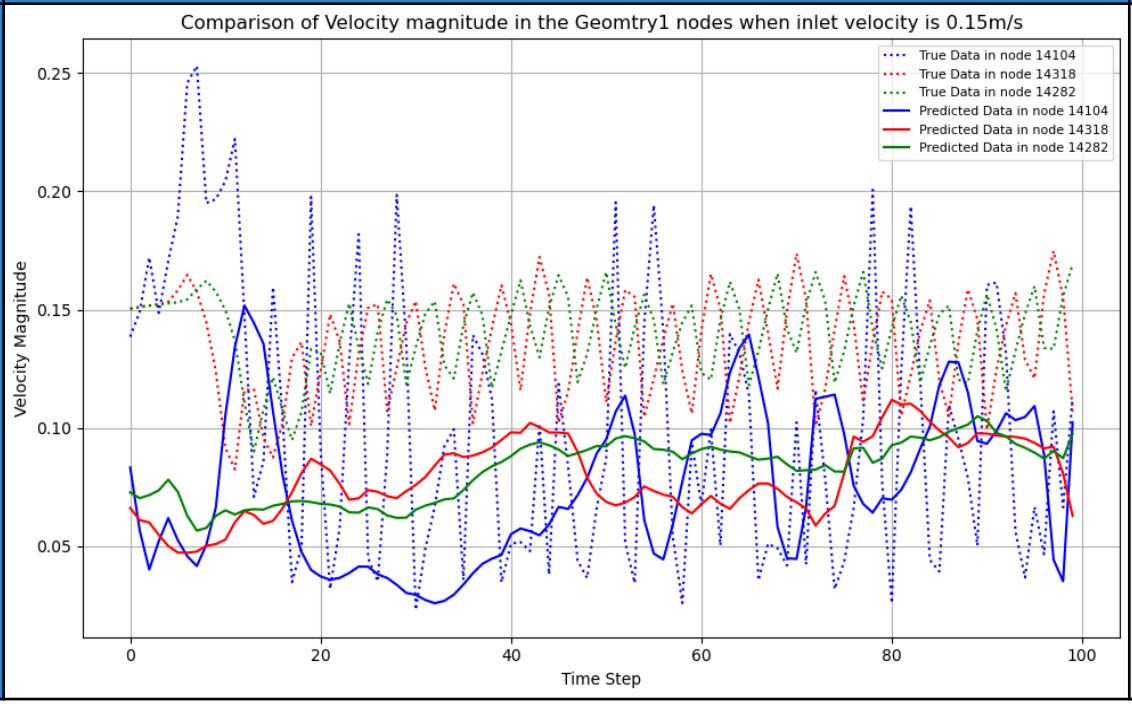
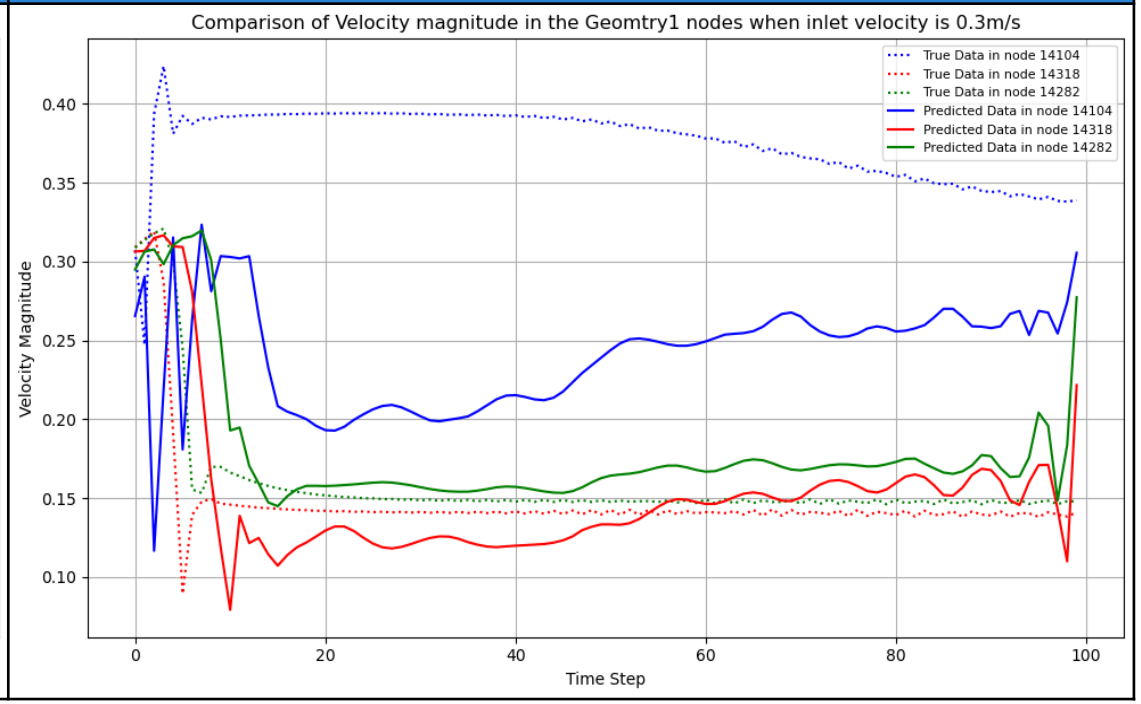
	Geometry2 – 0.15m/s	Geometry2 – 0.3m/s
Ground truth		
Prediction		

Geometry2 Results

- Under all evaluation scenarios, the model demonstrates a good ability to reproduce the overall flow pattern.
- Although the Kármán vortex street is not clearly observed before 50 time steps, the predicted flow closely resembles the reference solution thereafter.

Results

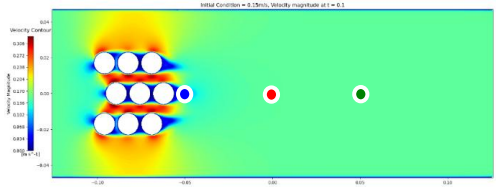


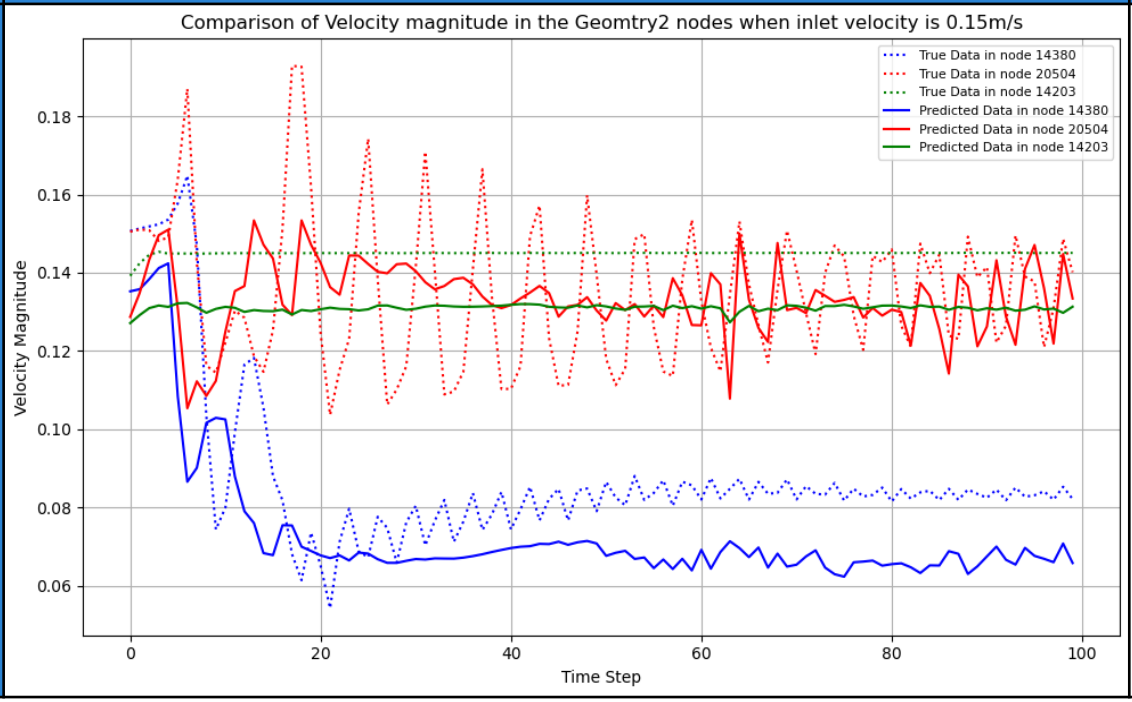
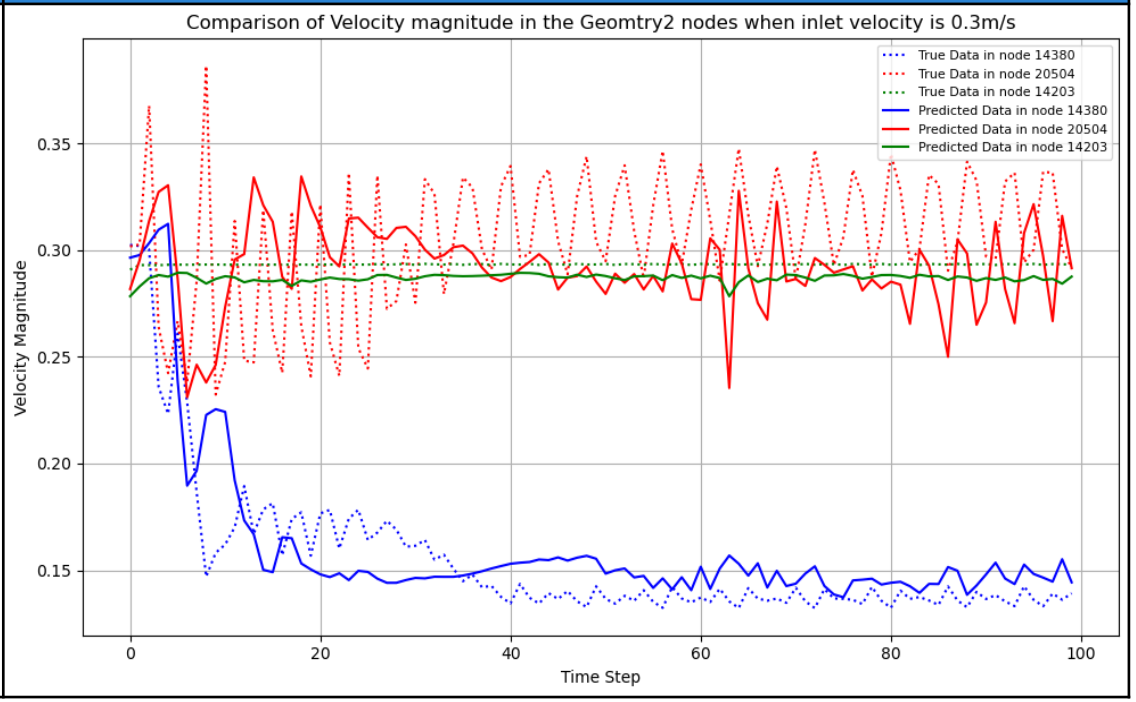
	Geometry1 – 0.15m/s	Geometry1 – 0.3m/s
Velocity distribution		

Geometry1 Results

- In addition, the velocity distribution over time was quantitatively examined at nodes located at the center of the geometry and at positions 50 mm to the left and right.
- Same with the previous results, the prediction at 0.15 m/s shows significant deviation from the actual velocity, while at 0.3 m/s, the predicted flow generally resembles the ground truth but still exhibits some discrepancies.

Results



	Geometry2 – 0.15m/s	Geometry2 – 0.3m/s
Velocity distribution		

Geometry2 Results

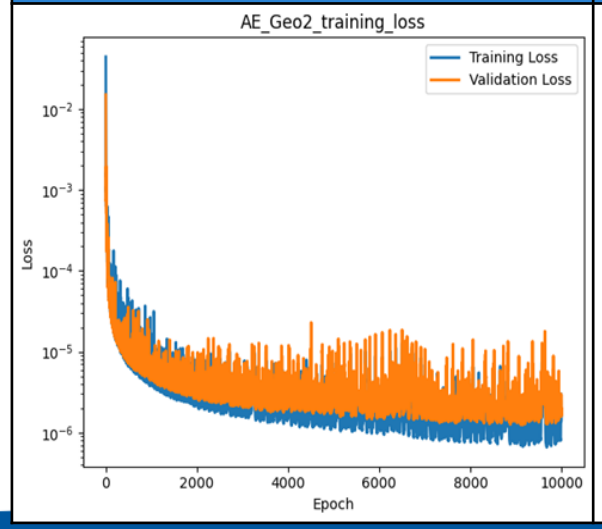
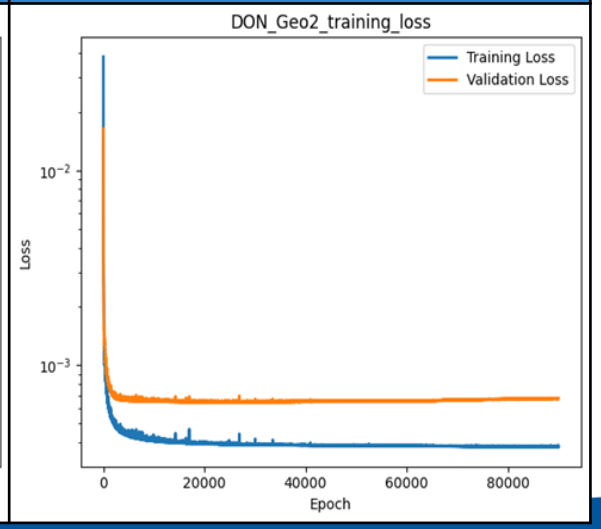
- In Geometry 2, the model successfully captured the overall flow behavior under both evaluation conditions.
- Although it did not accurately predict the magnitude of the velocity, the periodic flow pattern of the Kármán vortex was observed at all three nodes.

Results

Expansion to CFD acceleration model

- These results showed that satisfactory prediction performance can be achieved even with a limited amount of training data in complex geometries.
- To extend L-DeepONet as a surrogate model for CFD simulations, additional data were generated under the same conditions and used to further train the model.
  - → A total of 50 datasets were generated by varying the initial velocity from 0.01 m/s to 0.5 m/s.
  - → Of these, 40 were used as training data and 10 as validation data for training the L-DeepONet model.

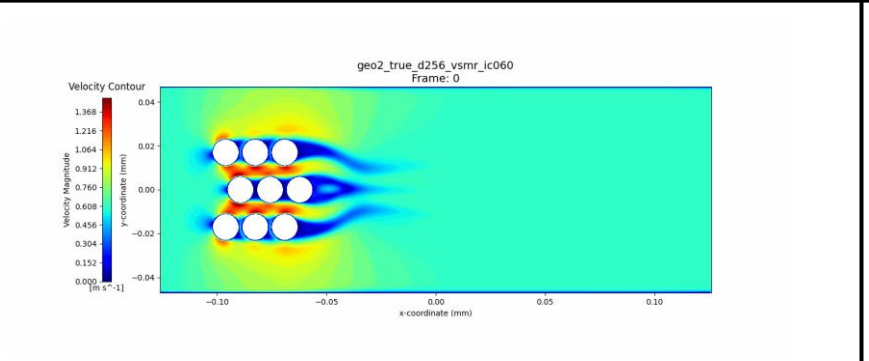
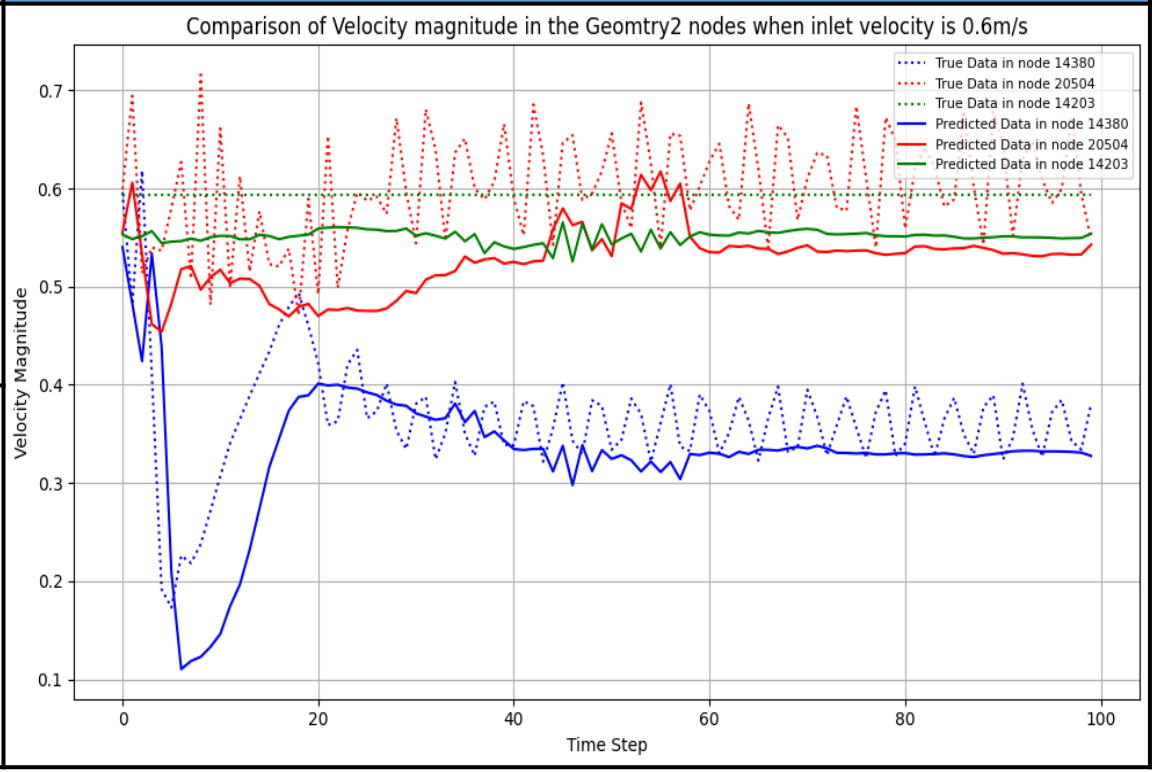
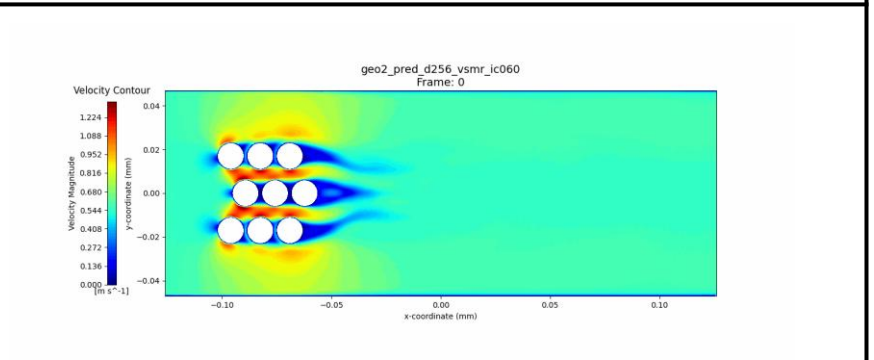
❖Loss graph

AutoEncoder Loss	DeepONet Loss
	

❖Training and Test Data condition

Training Data	40 samples (0.01~0.5 m/s)	
Validation Data	10 test samples (excluded from training)	
Test Data	Interpolation	0.4m/s
	Extrapolation	0.6m/s

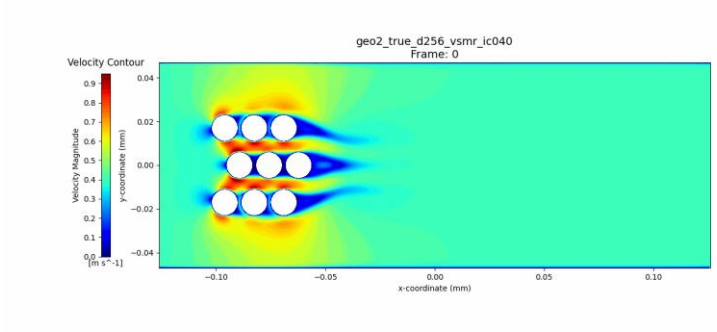
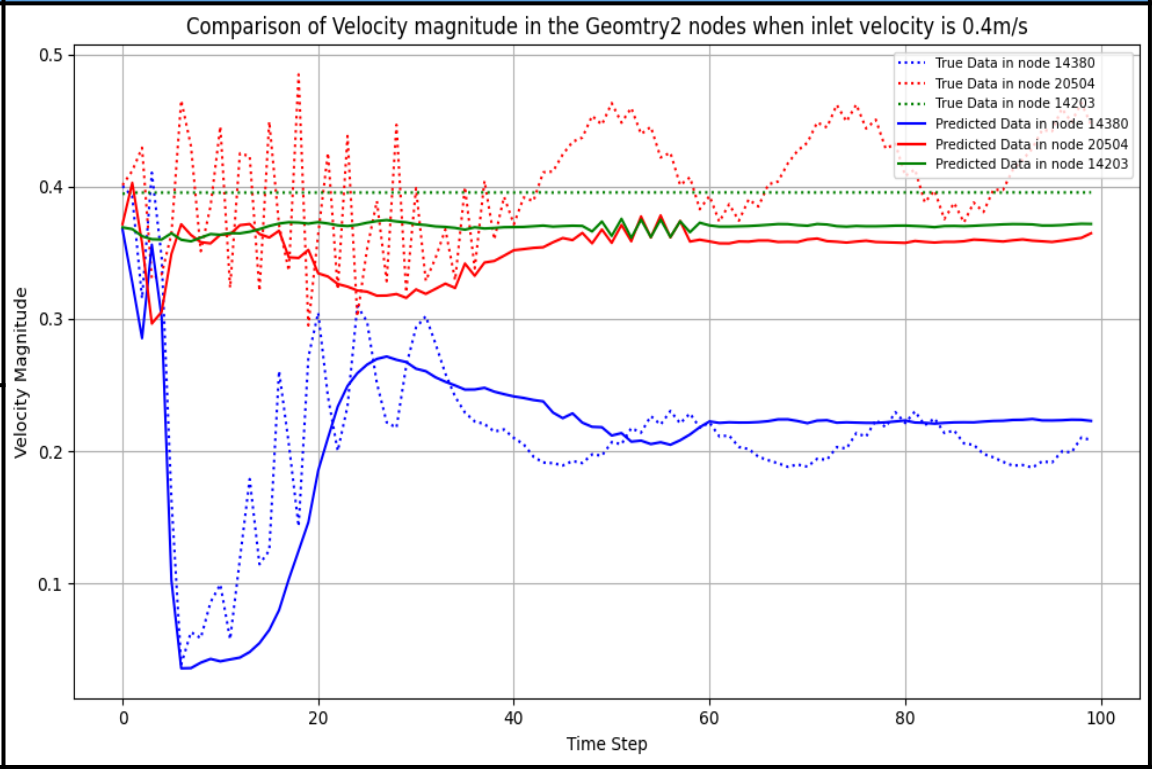
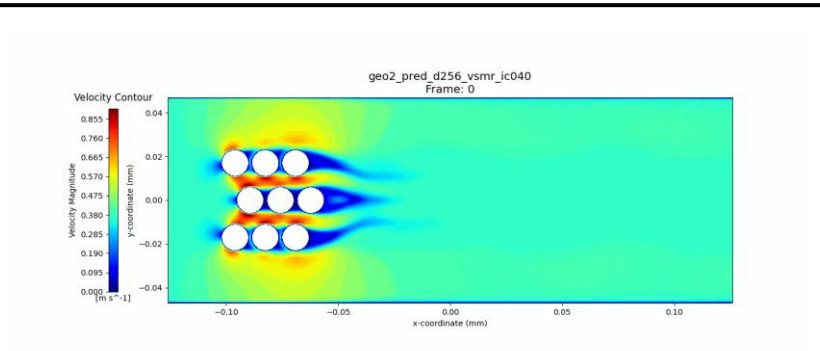
Results

	Geometry2 -0.6m/s	Velocity distribution – 0.6m/s
Ground truth		
Prediction		

Results when I.C 0.6m/s

- Unlike the previous results, the Kármán vortex in the cylinder wake is captured to some extent but gradually fades over time.
- However, the overall flow pattern is well predicted, and the mean velocity field is accurately represented.

Results

	Geometry2 -0.4m/s	Velocity distribution – 0.4m/s
Ground truth		
Prediction		

Results when I.C 0.4m/s

- As with the extrapolation results, the model shows good prediction of the overall flow.
- It briefly captures the periodic Kármán vortex pattern along the main flow, but the prediction quickly converges to the mean distribution.

Contents

Conclusion

Results Analysis

» Why did Geometry 2 exhibit better performance despite having a more complex shape than Geometry 1?

→ Because the training data samples in Geometry 2 are more similar to each other.

» Why did the representation of the Kármán vortex deteriorate despite the increase in training data?

✓ **Spectral Bias**

It refers to the tendency of neural networks to learn low-frequency features first and high-frequency features later.

→ Because of **spectral bias**, the autoencoder struggles to learn high-frequency features at local regions.

→ However, it can effectively learn the overall structure of the data.

→ Since DeepONet learned the flow field primarily based on the low-frequency components compressed by the autoencoder, it was able to accurately predict the macroscopic flow pattern rather than the strong periodicity of the Kármán vortex.

Conclusion

Conclusion and Future work

- The L-DeepONet model was applied to the geometry of a Helical Coil Steam Generator, and despite being trained on a limited dataset, it successfully accelerated CFD simulations even for some complex configurations.

  ❖ Inference Time
  - CFD simulations : **30 minutes** for each initial condition
  - L-DeepONet : **4s**

- As the training data increased, high-frequency features were smoothed out as expected, leading to a diminished representation of the Kármán vortex, although further performance improvement remains necessary.
- Future work will focus on extending the tailored surrogate model for SMR applications by incorporating a variety of geometries and boundary conditions like Geometry1 and under figures.

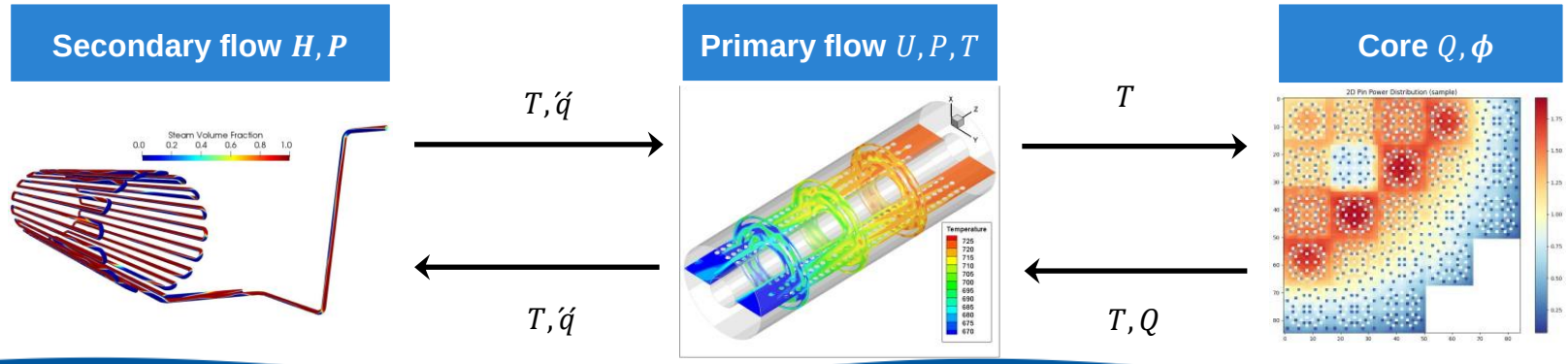


Figure from Yonggyun Yu (KAERI)

Acknowledgment

This work was supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MIST) (No. GTL24031-000) and the Nuclear Safety Research Program through the Korea Foundation of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (no. RS-2024-00403364).

V-SMR
Virtual-Small Modular Reactor

KAERI
한국원자력연구원
Korea Atomic Energy Research Institute

KoFONS
한국원자력안전재단
KOREA FOUNDATION OF NUCLEAR SAFETY

Thank you for listening!

msy246@jbnu.ac.kr

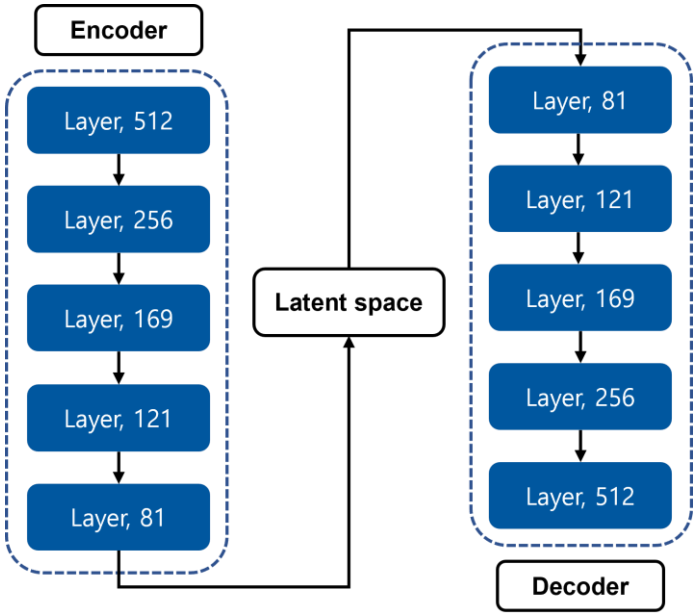
Appendix

L-DeepONet architecture and hyperparameter to learn 3 data

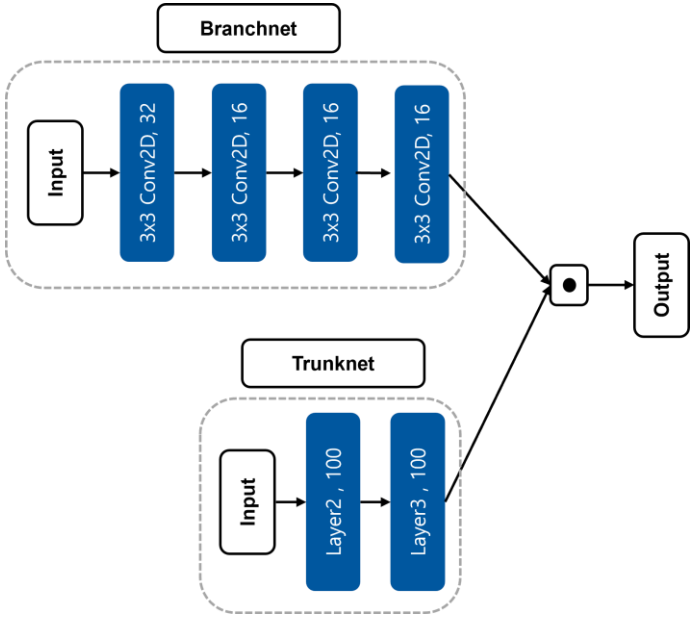
❖ Hyperparameters of L-DeepONet

- ❖ AutoEncoder(81 dimension)
  - Epoch : 10000
  - Batch size : 90
  - Learning weight : Adam(10e-03)
  - Activation function : ReLU
  - Take 1 hour to learn
- ❖ DeepONet
  - Epoch : 90000
  - Batch size : 3
  - Learning weight : Adam(10e-04)
  - Activation function : Sin
  - Take 5 minuites to learn

❖ AutoEncoder architecture



❖ DeepONet architecture



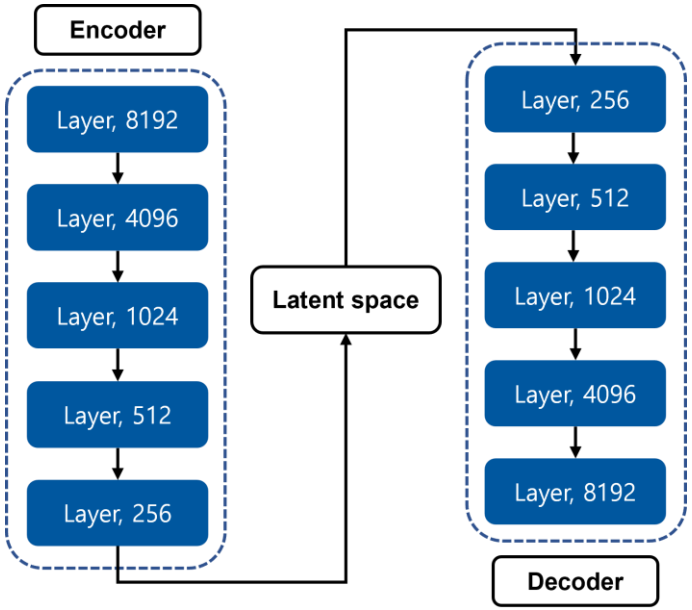
Appendix

L-DeepONet architecture and hyperparameter to learn 50 data

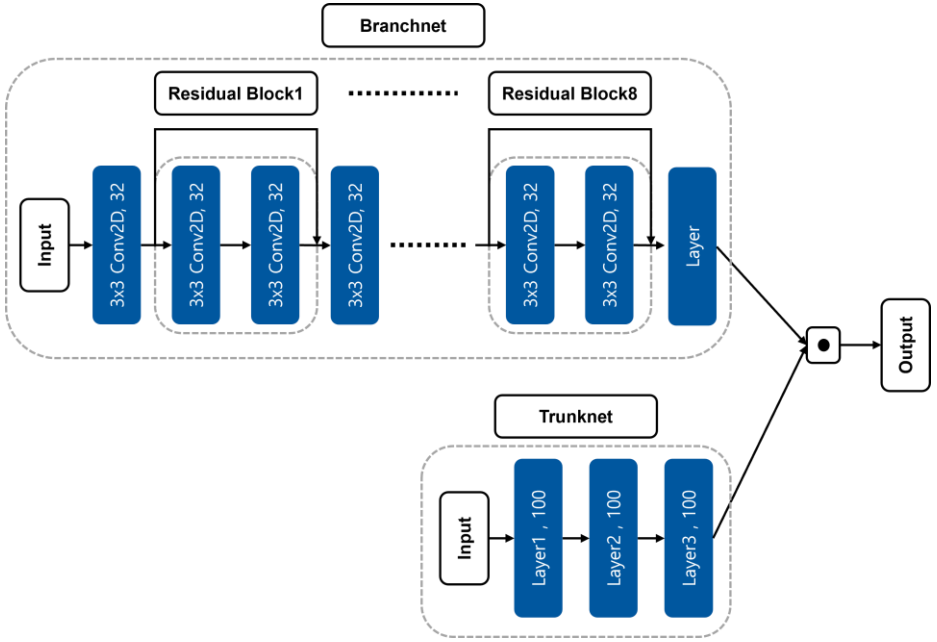
❖ Hyperparameters of L-DeepONet

- ❖ AutoEncoder(256 dimension)
  - Epoch : 10000
  - Batch size : 64
  - Learning weight : Adamw(10e-05)
  - Activation function : ReLU
  - Take 5 hours to learn
- ❖ DeepONet
  - Epoch : 90000
  - Batch size : 10
  - Learning weight : Adam(10e-04)
  - Activation function : ReLU
  - Take 15 minuites to learn

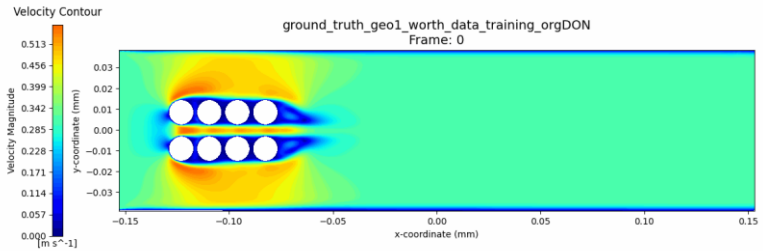
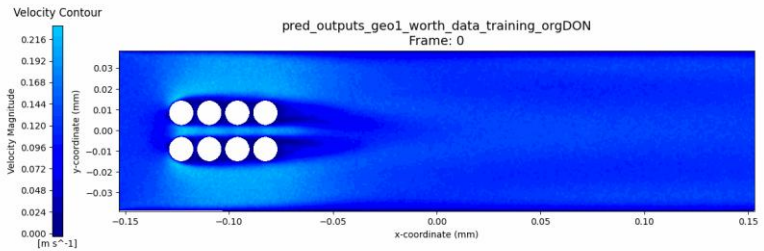
❖ AutoEncoder architecture



❖ DeepONet architecture



Results

	Ground Truth	Prediction
Geometry1 – 0.3m/s		

Orginal DeepONet results

- For performance comparison, a standard DeepONet was also trained on the same dataset using the Geometry 1 configuration.0.3m/s
- Under the 0.3 m/s condition, while the overall flow pattern appeared similar, there was a significant discrepancy in velocity magnitude.
- The predicted flow field showed similar patterns but underestimated the velocity magnitude.