CFD Assisted Deep Learning Approach for Predicting Thermal Flows for Hotspot Mitigation in Data Center Racks

By

Gulzar Ali

Master of Science in Computational Science & Engineering

School of Interdisciplinary Engineering and Sciences

National University of Sciences and Technology

Supervisor:

Dr. Absaar ul Jabbar
Assistant Professor
SINES, NUST

GEC Members:

Dr. Ammar Mushtaq
Associate Professor
SINES, NUST

Dr. Muhammad Irfan Zafar
Assistant Professor
SMME, NUST

Outline

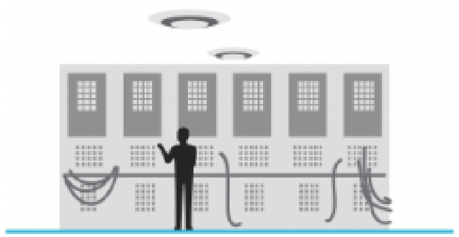
Introduction

“Data centers and supercomputers are the backbone of modern digital infrastructure, enabling innovation and powering critical operations across industries.”

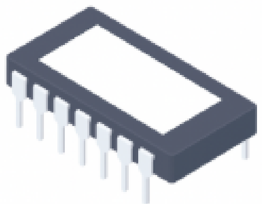
Both rely on ***efficient cooling systems*** and ***power management*** to handle intensive workloads and ensure reliability.

References: https://www.vecteezy.com/vector-art/22234208-isometric-datacenter-server-room-concept-server-room-data-center-vector-illustration

Introduction

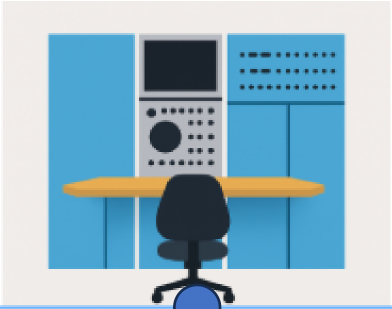


1946
ENIAC (Electronic Numerical Integrator And Computer) was the **first electronic general-purpose computer.**

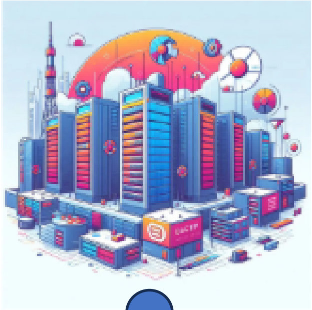


1971
The **Intel 4004** is a 4-bit central processing unit (CPU) released by Intel Corporation and it was the **first commercially available microprocessor.**

1981
The **IBM Personal Computer**, commonly known as the IBM PC, is the original version of the IBM PC compatible hardware platform.

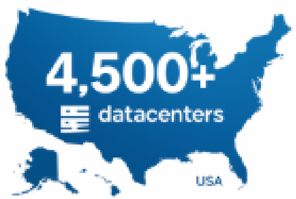


Early 1990s
Microcomputers (now called "servers") started to find their places in the old computer rooms and were being called "datacenters".

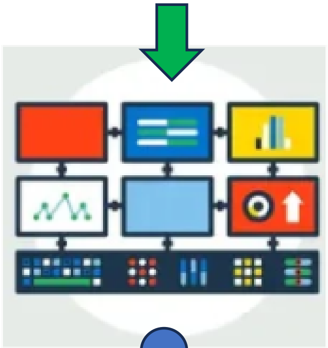


Mid 1990s
The **boom of datacenters** came during the dot-com bubble. Companies needed fast Internet connectivity and nonstop operation to deploy systems and establish a presence on the Internet.

2013
Google invested $7.35 billion in its Internet infrastructure. This spending was driven by an expansion of Google's global data center network. It represented the largest construction effort in the history of the datacenter.



2015
Over **5.75 million new servers** are deployed every year. There are an estimated **4,500+ datacenters** in the U.S. alone. To meet the growing demand of new applications and services, servers need to be deployed at an increasingly faster pace and larger number.

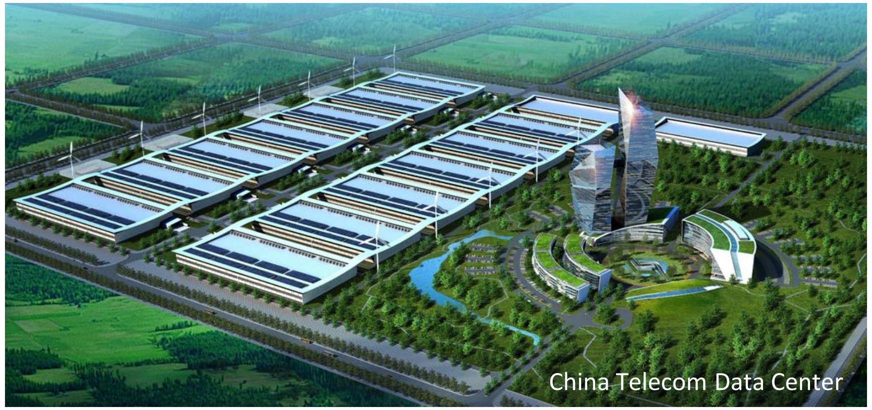


2018
Google began using **DeepMind AI** to autonomously control datacenter cooling. Resulted in a **40% reduction in energy used for cooling.**

Introduction

Some of the biggest datacenters and their energy consumptions:

Name	Location	Power Consumption
China Telecom Data Center	Hohhot, Inner Mongolia, China	150 MW
CWL1 Data Center	Newport, Wales, UK	148 MW
The Citadel Campus (when fully-built out)	Tahoe Reno, Nevada, USA	650 MW
Apple Mesa Datacenter	Mesa, Arizona, USA	50 MW
Lakeside Technology Center	Chicago, Illinois, USA	100 MW


China Telecom Data Center

Global data center market size was valued at *USD 219.23 billion* in ‘2023’ and is projected to USD *584.86 billion* by ‘2032’

References:
https://www.nxtra.in/blog/key-features-of-the-worlds-largest-data-center
https://brightlio.com/data-center-stats/
https://www.fortunebusinessinsights.com/data-center-market-109851

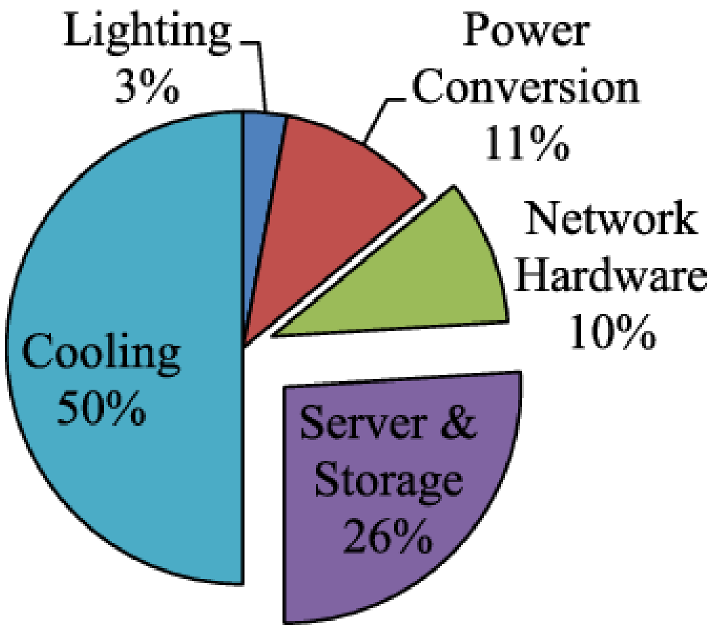
Introduction

Data centers consumed *460TWh* in 2022 and could rise to more than *1,000TWh* by 2026 in a worst-case scenario.
(IEA Annual Electricity Report)

IT Equipment:
Servers, storage, and networking devices typically consume ~30-40% of the total energy.

Cooling Systems:
Responsible for ~40-50% of energy use, ensuring optimal operating temperatures for IT equipment.

Power Infrastructure:
Includes UPS, power distribution units (PDUs), and transformers, accounting for ~10-20% of total energy use.



Energy Consumption in data Center

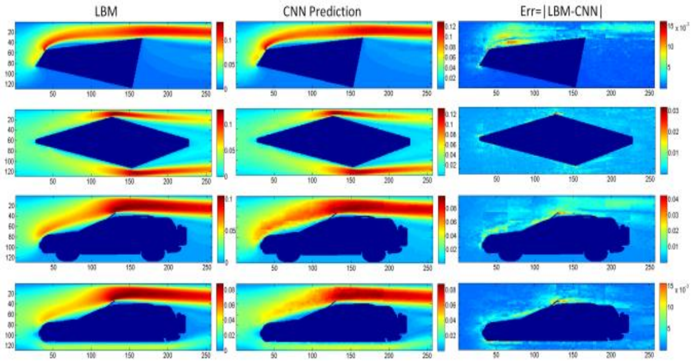
Introduction

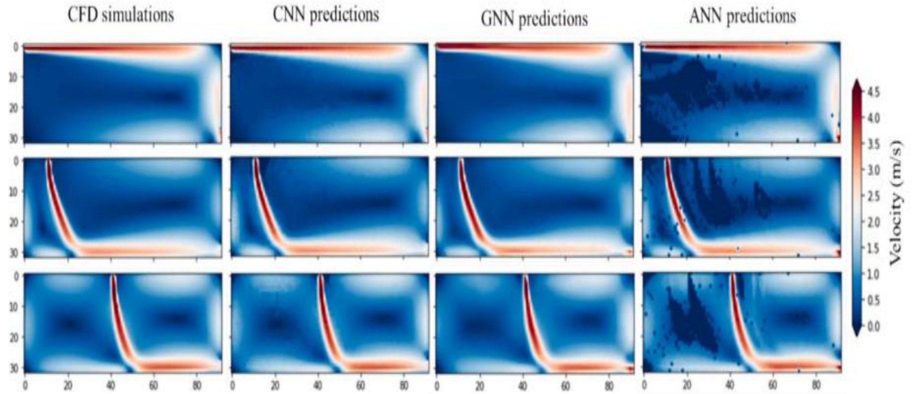
- **Conventional CFD**
  - Numerical Methods (FVM / FDM / FEM)
  - High Accuracy and Physical Consistency
  - High Computational Cost
  - Iterative Solver (Requires experts for monitoring)

- **CFD with AI**
  - Simulation Acceleration
  - Rapid Design Optimization
  - Improve the Accuracy of CFD Solvers
  - Flow Field Reconstruction and Automation

Literature Review

22nd ACM SIGKDD | Conference Paper (2016) **<u>Convolutional Neural Networks for Steady Flow Approximation</u>** XiaoXiao Guo, Wei Li, Francesco Iorio, Hao Zhu, Simon Hu	IBPSA | Conference Paper (2023) **<u>Convolutional neural networks-based surrogate model for fast computational fluid dynamics simulations of indoor airflow distribution</u>** Giovanni Calzolari, Wei Liu
• Predicts steady flow around obstacles using CNN • Maps SDF to get pressure and velocity fields. • This study further extended to work with 3D CNNs. • Does not map inlet conditions for investigating flow at different wind speed and directions • Max Absolute Error ~ 0.04	• Compared three surrogate models CNN, GNN and ANN to predict air flow in closed environment for different position of inlet velocity • CNN demonstrates best performance among all models





Literature Review

ELSEVIER | Building and Environment (2023)

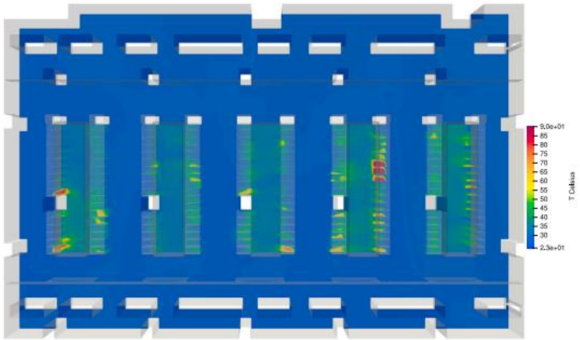
An open-source and experimentally guided CFD strategy for predicting air distribution in data centers with air cooling

Wei Liu, Song Lian, Xin Fang, Zhenyu Shang, Hao Wu, Hao Zhu, Simon Hu

Pure CFD gives Accurate

- Datacenter cooling design
- Velocity and Temperature prediction

Pure CFD lacks real-time insights

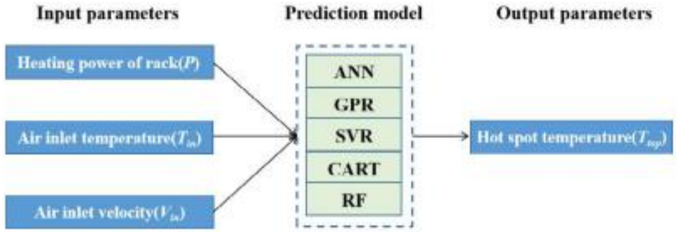


ELSEVIER | Building Simulation (2023)

Hot spot temperature prediction and operating parameter estimation of racks in data center using machine learning algorithms based on simulation data

Xianzhong Chen, Rang Tu, Ming Li, Xu Yang, Kun Jia

- Predicts maximum temperature (hotspot) in the rack based on inlet conditions
- Does not locate the position of hotspots



Problem Statement

Overcome the computational limitations of traditional CFD and Support rapid design-space exploration for thermal optimization in Datacenter Racks.

Key Challenges:

- ❑ Temperature sensors used in Racks doesn’t provide full thermal field
- ❑ 3D CFD Simulations are very expensive for dataset generation
- ❑ Training for large domains requires powerful GPUs

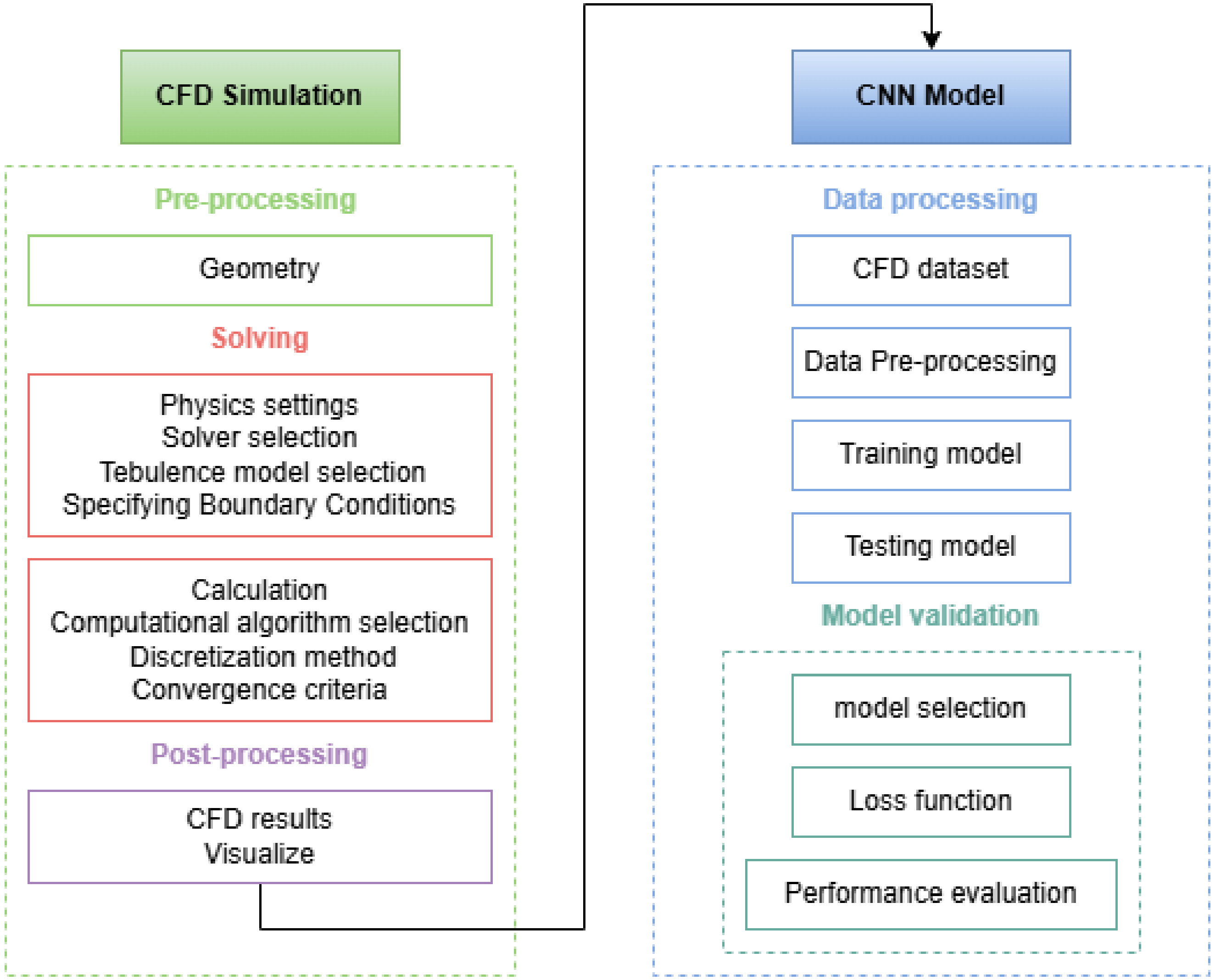
Proposed Solution

Develop a fast, accurate, and CFD assisted deep-learning framework capable of real-time thermal flow-field prediction and fast design-space exploration in data-center Racks.

Key Points:

- ❑ Train deep neural network on CFD generated data
- ❑ Predicts thermal flow fields instantly
- ❑ Enables proactive, efficient, and cost-effective thermal management
- ❑ Supports early-stage design and rapid exploration of large design spaces

Methodology

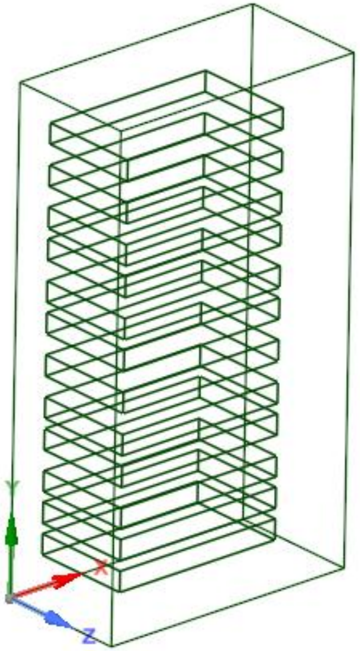


Methodology

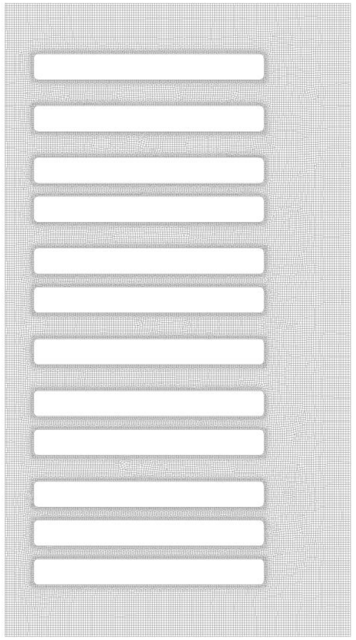
CFD Simulation Setup

Geometry

- A 2D slice of the rack section was extracted for CFD analysis.
- Model was validated using field data from data center located in Beijing, cross-verified in published studies.



3D Rack Model



2D Model

3D Server

Dimensions:
Rack: **1.2**(L) x **0.6**(W) x **2.2**(H)
Servers: **0.8**(l) x **0.46**(w) x **0.09**(h)

References:

[1] Yuan X, Zhou X, Liu J, et al. (2019). Experimental and numerical investigation of an airflow management system in data center with lower-side terminal baffles for servers. Building and Environment, 155: 308–319.

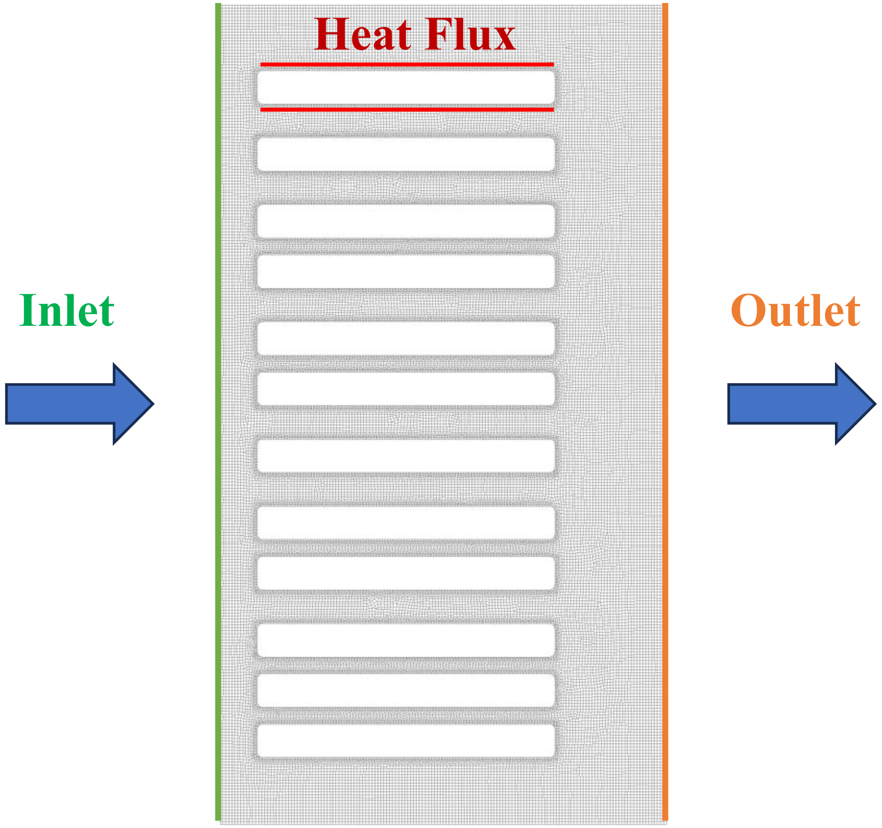
[2] Yuan X, Xu X, Liu J, et al. (2020). Improvement in airflow and temperature distribution with an in-rack UFAD system at a high-density data center. Building and Environment, 168: 106495.

Methodology

CFD Simulation Setup

Type	Description	Value
Boundary Conditions	Inlet	Velocity Inlet
	Outlet	Pressure Outlet
	Top and Bottom Surface	Constant Heat Flux

Type	Value
Solver Type	Pressure-based
Study Type	Steady State
Turbulence Model	Standard **k–ε**
Near-wall Treatment	Standard wall functions

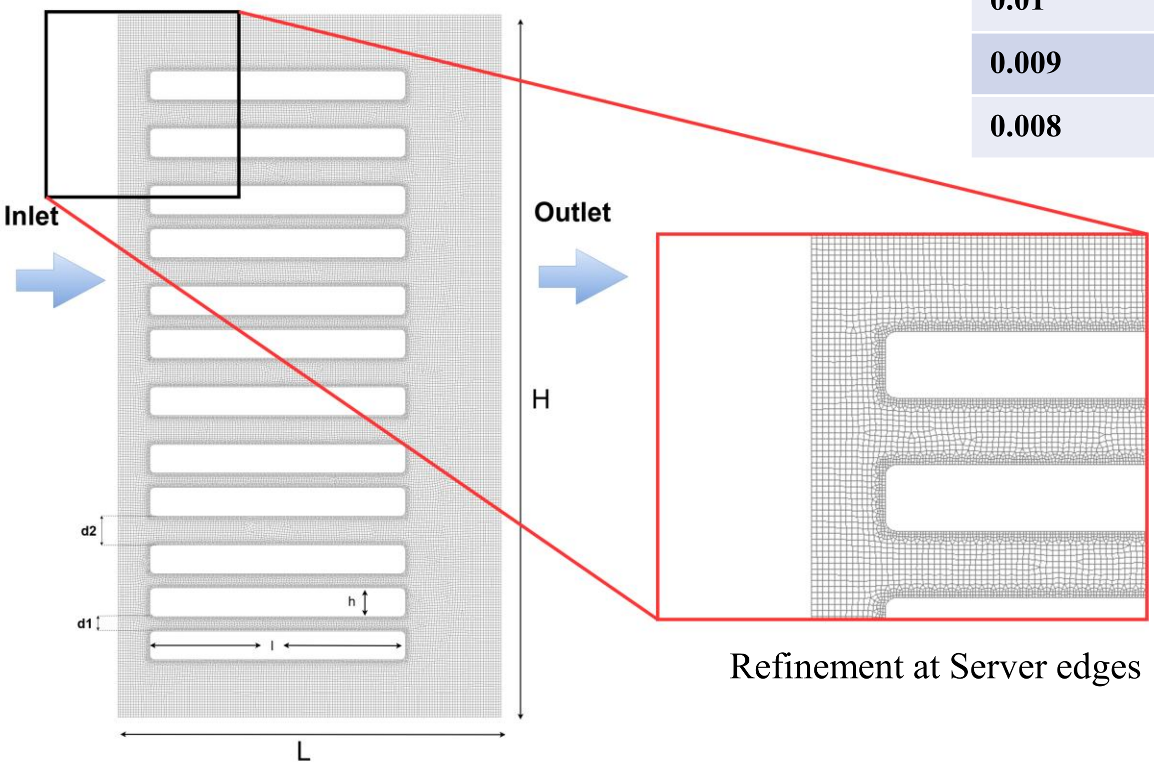


Methodology

CFD Simulation Setup

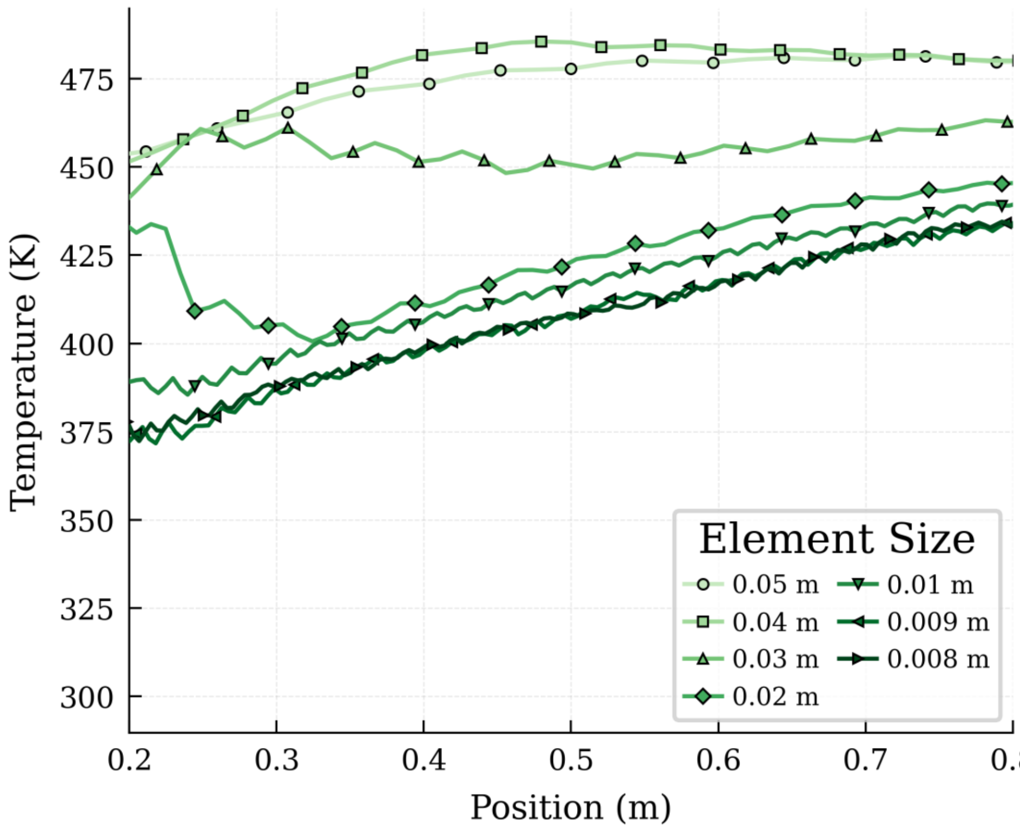
Element Size (m)	Number of Elements
0.04	3470
0.03	5479
0.02	9894
0.01	30403
0.009	35924
0.008	43836

Final Mesh



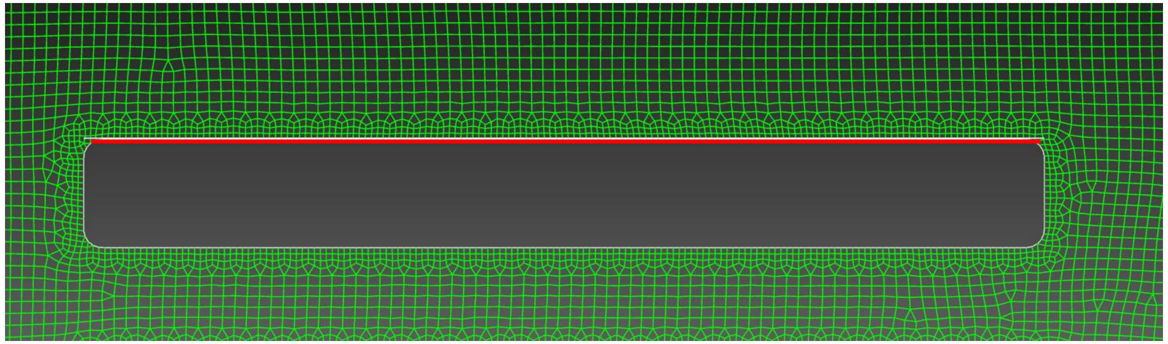
Refinement at Server edges

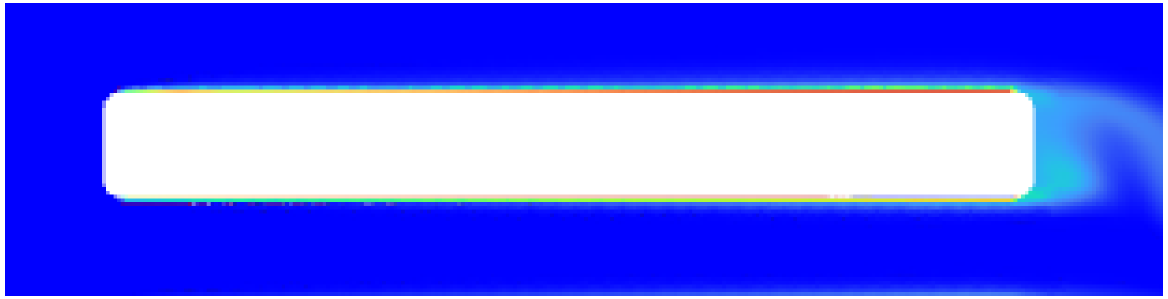
Mesh Independence

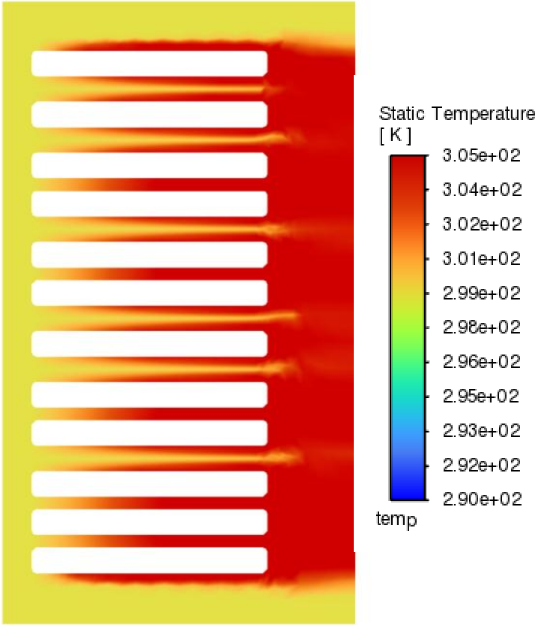


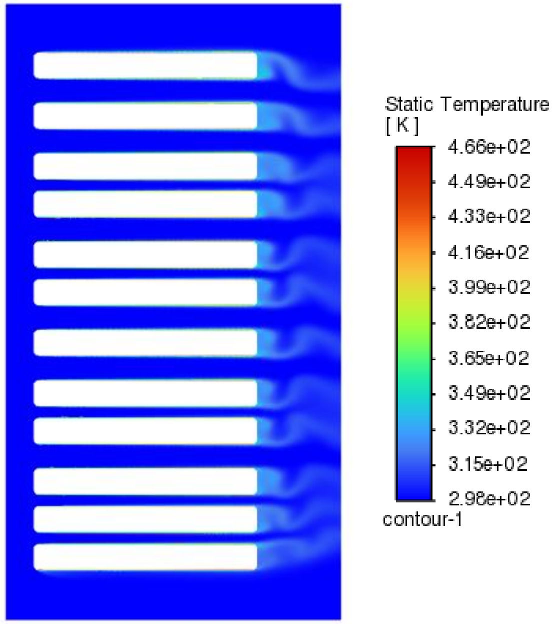
Methodology

CFD Simulation Setup









Methodology

CFD Simulations Setup

Parametric Simulations

- Fluent Parametric Simulations were used to built comprehensive dataset
- Two dataset were generated
  - Phase 1 (Uniform Server Power Distribution)
  - Phase 2 (Non-uniform server Power Distribution)

Phase II: combinations
Temp = 3
Vel = 3
Server Power (**P**) = 4
Servers (**S**)= 12
Combinations = **Temp** x **Vel** x $\mathbf{P}^{\mathbf{S}}$
= 150,994,994

?

Parameter	Total Values	Range
Velocity (m/s)	10	1.75 to 2.65
Temperature (K)	10	290 to 299
Server Power (W)	20	416 to 1666

PHASE I (2000 Simulations)

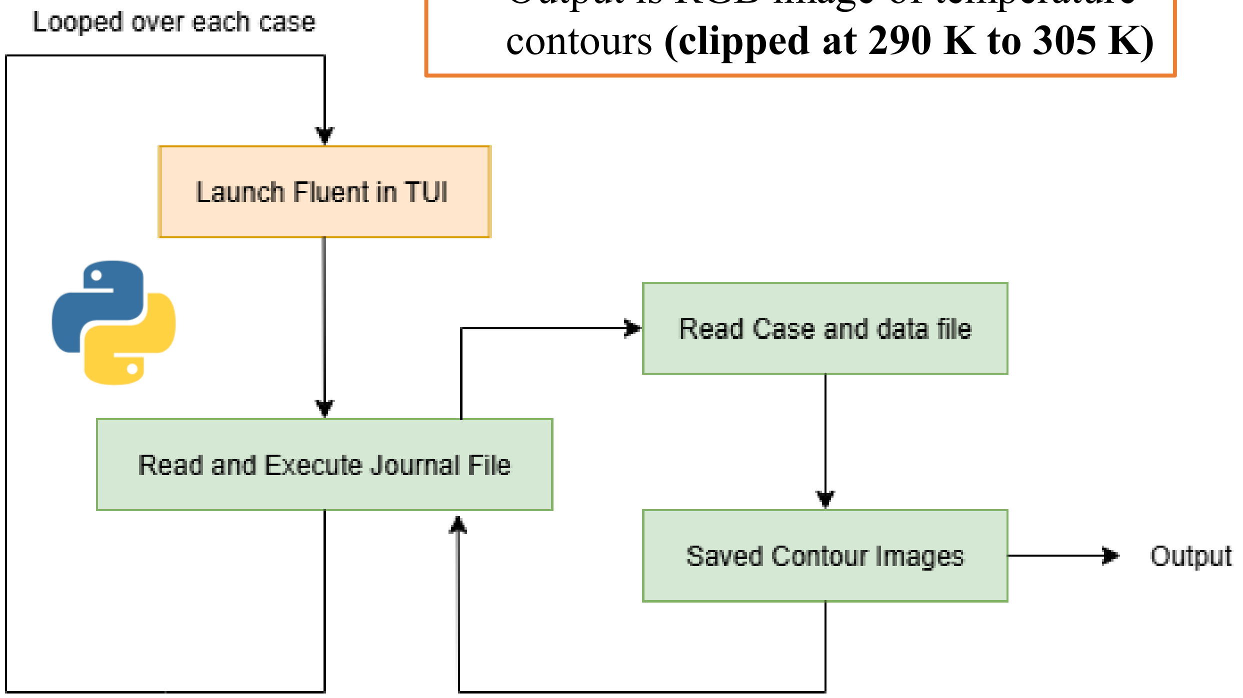
Parameter	Total Values	Range
Velocity (m/s)	**3**	1.75 to 2.65
Temperature (K)	**3**	290 to 299
12 x Server Power (W)	**4**	**0 to 1500**

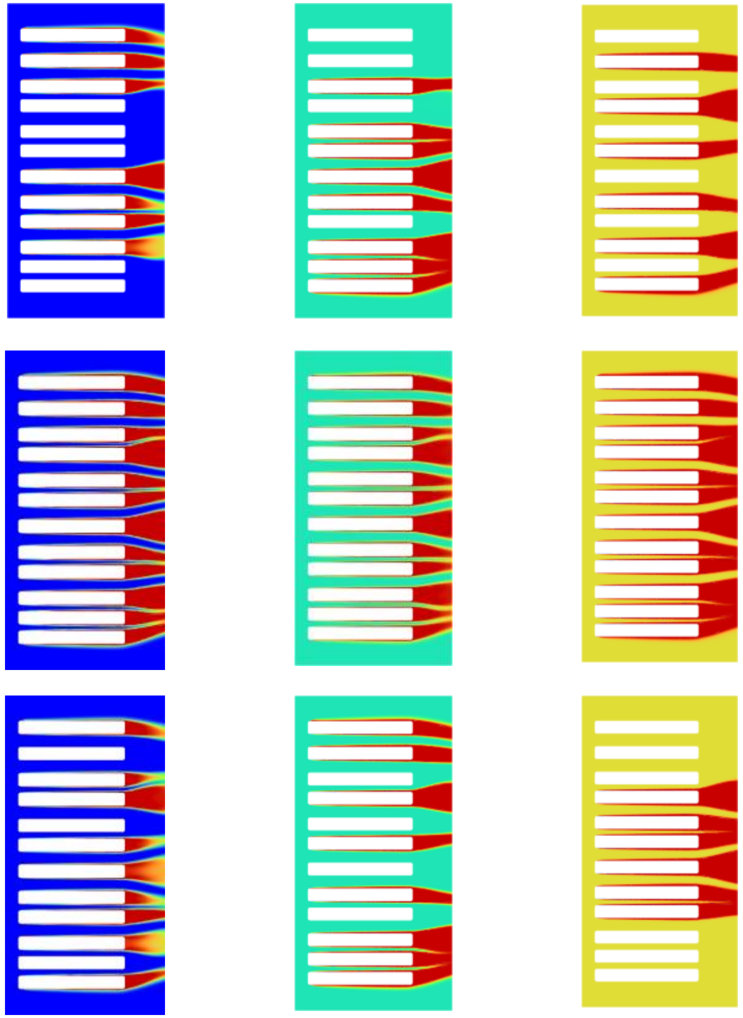
PHASE II (2590 Simulations)

Methodology

Parametric Simulations

- Python code for data extraction using custom journal files
- Output is RGB image of temperature contours **(clipped at 290 K to 305 K)**





Methodology

Data Preprocessing

- ❑ Meta data consist of **inlet velocity , server power** and **inlet temperature** mapped to true **RGB Images**
- ❑ Images are resized to 128 by 256
- ❑ All scalars are normalized to [-1, 1], using min-max normalization

$$x_{norm} = 2\left(\frac{x - x_{min}}{x_{max} - x_{min}}\right) - 1$$

- ❑ Image Pixel values are scaled to [0, 1] range

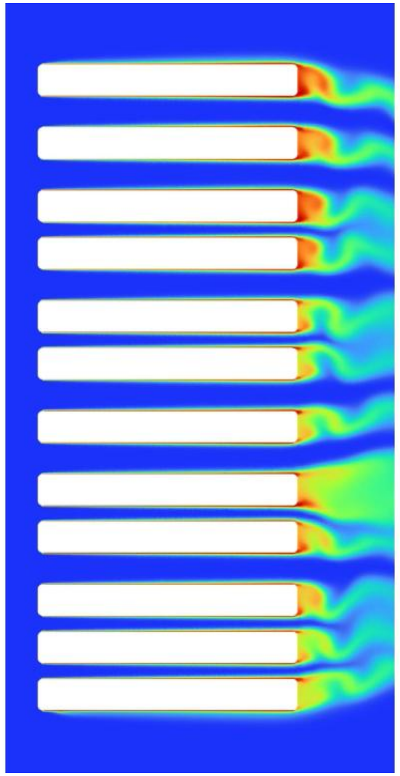
$$x_{norm} = \frac{x}{255}$$

Methodology

Parametric Simulations

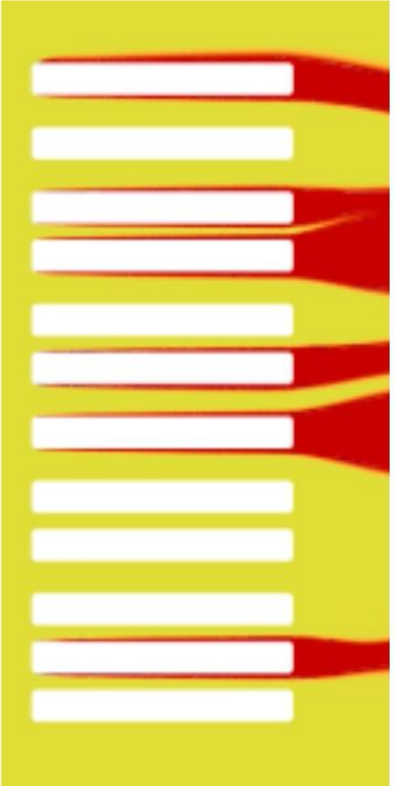
Temperature Contours

Phase I



Generated Dataset

Phase II



Methodology

Model Architecture

Three Models were tested on the generated Phase 1 data set

Model 1 (Simple Deconv Decoder)

Layer (type)	Output Shape	Param #
Linear-1	[-1, 1024]	4,096
ReLU-2	[-1, 1024]	0
Linear-3	[-1, 2048]	2,099,200
ReLU-4	[-1, 2048]	0
Dropout-5	[-1, 2048]	0
ConvTranspose2d-6	[-1, 128, 8, 4]	524,416
ReLU-7	[-1, 128, 8, 4]	0
ConvTranspose2d-8	[-1, 64, 16, 8]	131,136
ReLU-9	[-1, 64, 16, 8]	0
ConvTranspose2d-10	[-1, 32, 32, 16]	32,800
ReLU-11	[-1, 32, 32, 16]	0
ConvTranspose2d-12	[-1, 16, 64, 32]	8,208
ReLU-13	[-1, 16, 64, 32]	0
ConvTranspose2d-14	[-1, 8, 128, 64]	2,056
ReLU-15	[-1, 8, 128, 64]	0
ConvTranspose2d-16	[-1, 3, 256, 128]	387
ReLU-17	[-1, 3, 256, 128]	0
Conv2d-18	[-1, 3, 256, 128]	84
Sigmoid-19	[-1, 3, 256, 128]	0

Total params: 2,802,383
...
Forward/backward pass size (MB): 5.00
Params size (MB): 10.69
Estimated Total Size (MB): 15.69

Terms Used:

- Linear (Fully connected layer)
- BatchNorm (Normalizes activations)
- ReLU (Nonlinear activation)
- Upsample (Increases spatial resolution)
- ConvTranspose2d (Learnable upsampling)
- Sigmoid (Squashes values to (0,1))

Methodology

Model Architecture

Three Models were tested on the generated Phase 1 data set

Model 2 (Residual Deconv Decoder)

Layer	Output Shape	Param #
Linear-1	[B, 1024]	4,096
ReLU-2	[B, 1024]	0
Linear-3	[B, 2048]	2,099,200
ReLU-4	[B, 2048]	0
UpsampleBlock1-5	[B, 128, 8, 4]	743,424
UpsampleBlock2-6	[B, 64, 16, 8]	185,472
UpsampleBlock3-7	[B, 32, 32, 16]	46,368
UpsampleBlock4-8	[B, 16, 64, 32]	11,664
UpsampleBlock5-9	[B, 8, 128, 64]	2,952
UpsampleBlock6-10	[B, 4, 256, 128]	768
Conv2d-11	[B, 3, 256, 128]	111
Sigmoid-12	[B, 3, 256, 128]	0
Total Parameters		3,093,031

Forward/backward pass size (MB): 27.14
Params size (MB): 11.03
Estimated Total Size (MB): 38.17

```python
class ResidualBlock(nn.Module):
    def __init__(self, channels):
        super().__init__()
        self.block = nn.Sequential(
            nn.Conv2d(channels, channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(channels),
            nn.ReLU(inplace=True),
            nn.Conv2d(channels, channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(channels)
        )
        self.relu = nn.ReLU(inplace=True)

    def forward(self, x):
        return self.relu(x + self.block(x))
```

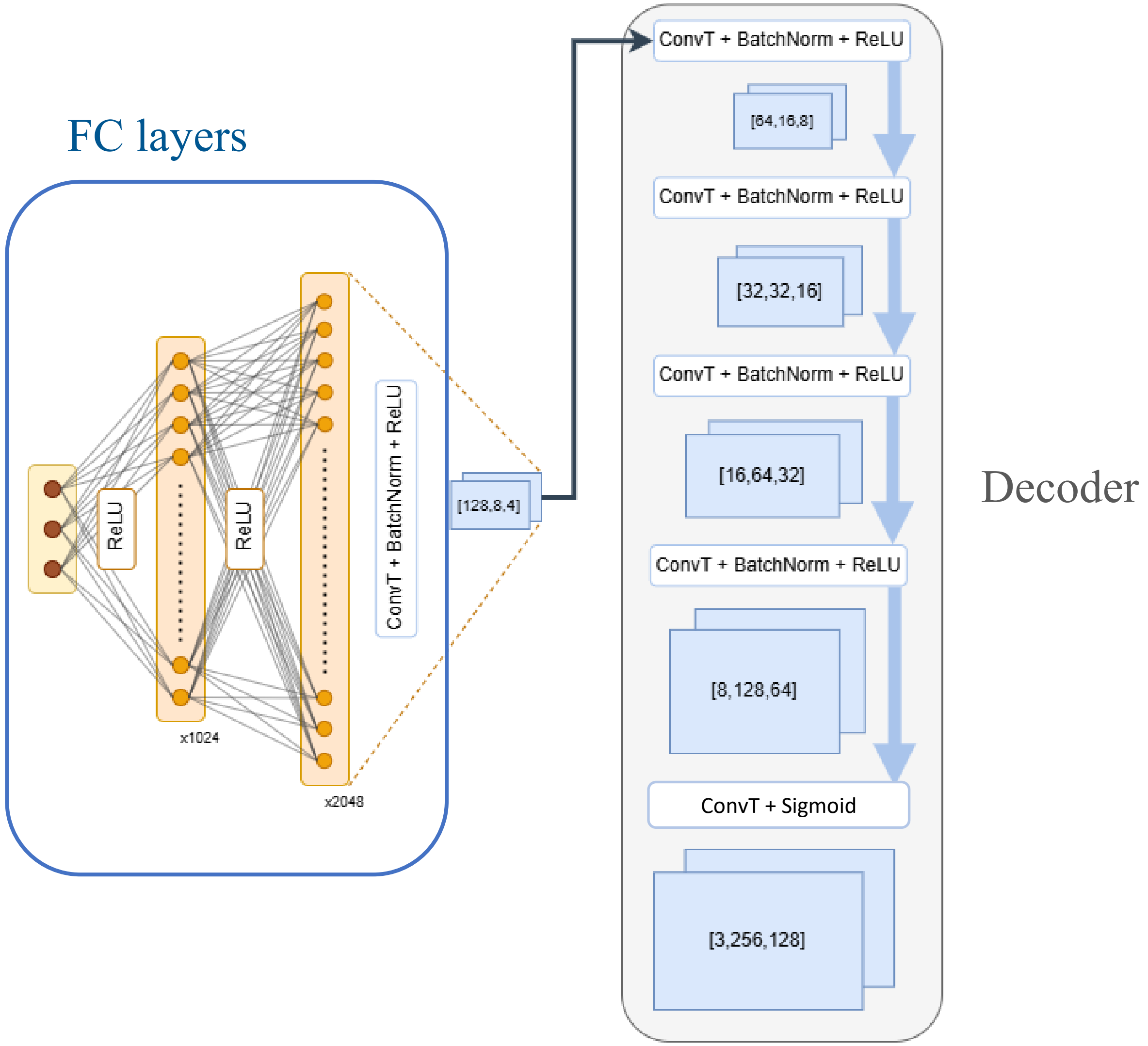
```python
class UpsampleBlock(nn.Module):
    def __init__(self, in_channels, out_channels):
        super().__init__()
        self.upsample = nn.Sequential(
            nn.Upsample(scale_factor=2, mode='nearest'),
            nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1),
            nn.BatchNorm2d(out_channels),
            nn.ReLU(inplace=True),
            ResidualBlock(out_channels)
        )

    def forward(self, x):
        return self.upsample(x)
```

Methodology

Model Architecture

Model 3 (Normalized Deconv Decoder)

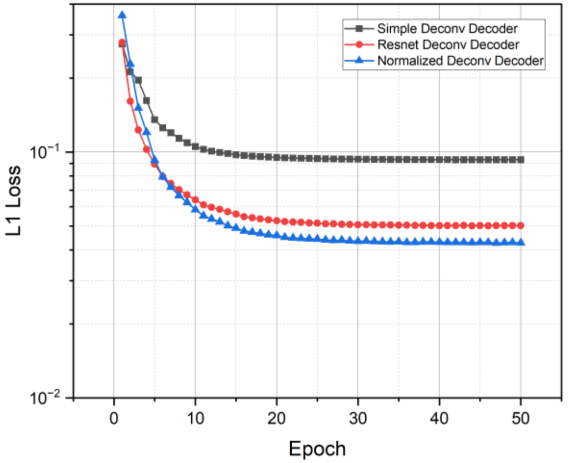
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                [-1, 1024]           4,096
              ReLU-2                [-1, 1024]               0
            Linear-3                [-1, 2048]       2,099,200
              ReLU-4                [-1, 2048]               0
   ConvTranspose2d-5           [-1, 128, 8, 4]         524,416
       BatchNorm2d-6           [-1, 128, 8, 4]             256
              ReLU-7           [-1, 128, 8, 4]               0
   ConvTranspose2d-8           [-1, 64, 16, 8]         131,136
       BatchNorm2d-9           [-1, 64, 16, 8]             128
             ReLU-10           [-1, 64, 16, 8]               0
  ConvTranspose2d-11          [-1, 32, 32, 16]          32,800
      BatchNorm2d-12          [-1, 32, 32, 16]              64
             ReLU-13          [-1, 32, 32, 16]               0
  ConvTranspose2d-14          [-1, 16, 64, 32]           8,208
      BatchNorm2d-15          [-1, 16, 64, 32]              32
             ReLU-16          [-1, 16, 64, 32]               0
  ConvTranspose2d-17          [-1, 8, 128, 64]           2,056
      BatchNorm2d-18          [-1, 8, 128, 64]              16
             ReLU-19          [-1, 8, 128, 64]               0
  ConvTranspose2d-20         [-1, 3, 256, 128]             387
          Sigmoid-21         [-1, 3, 256, 128]               0
...
Forward/backward pass size (MB): 4.45
Params size (MB): 10.69
Estimated Total Size (MB): 15.14
```



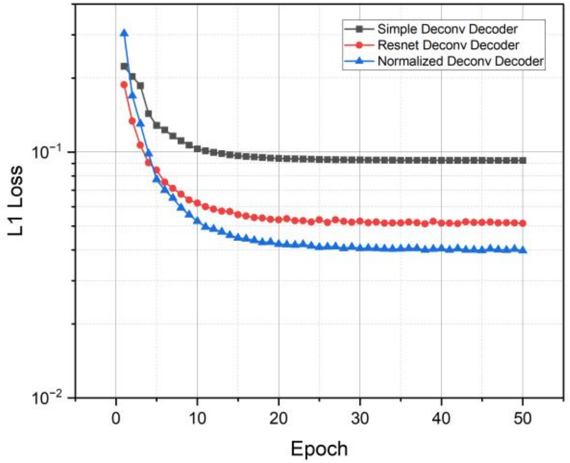
Methodology

Model Training

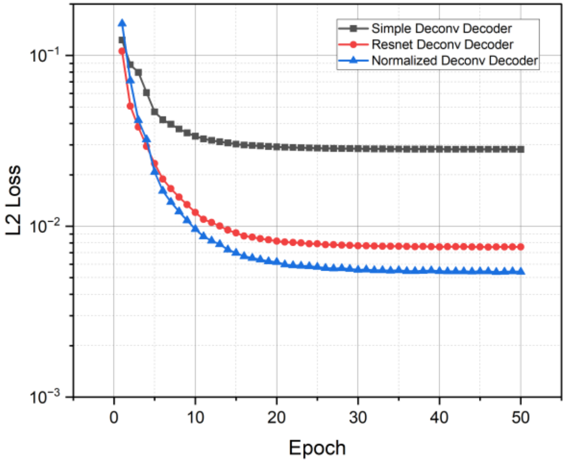
Training Loss

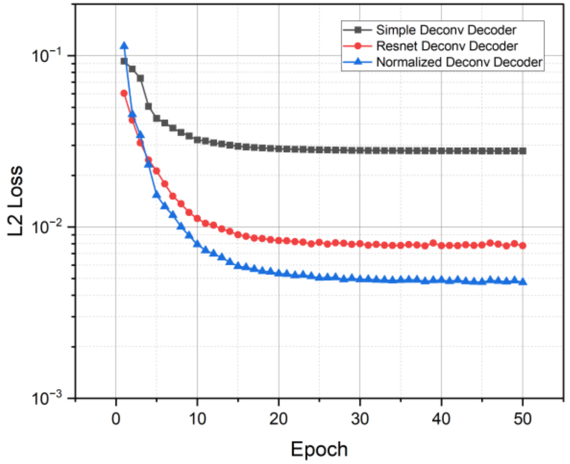


Validation Loss

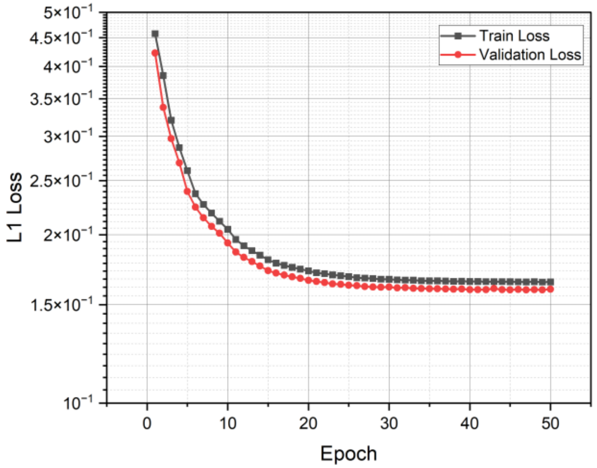


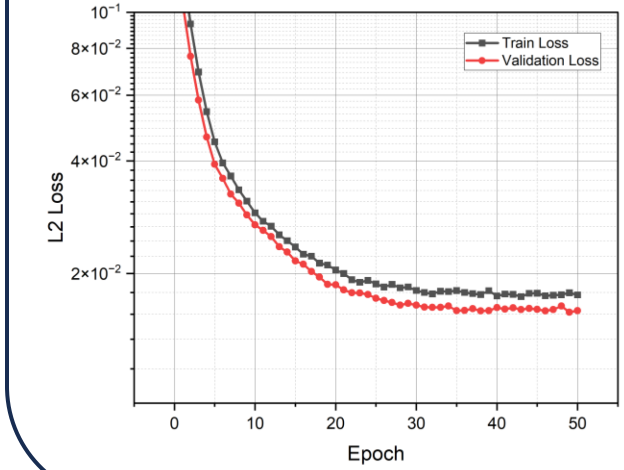
PHASE I





PHASE II





Methodology

Model Training

Training Configurations

Parameter	PHASE I	PHASE II
Optimizer	Adam	Adam
Learning Rate	1×10^{-4}	1×10^{-3}
Learning Rate Scheduler	StepLR (decay every 5 epochs)	No
Batch Size	2	8
Epochs	Early Stopping (patience = 5 epochs)	50
Gradient Clipping	Max norm = 1	Max norm = 1

During Phase II, Model is trained step wise;

Step 1: trained on uniform powers and extreme conditions

Step 2: trained using patterns like (increasing and decreasing, alternative on/off, upper, middle and lower servers working)

Step 3: the model was trained on random dataset

For Final Model after Hyper Parameteric Study:

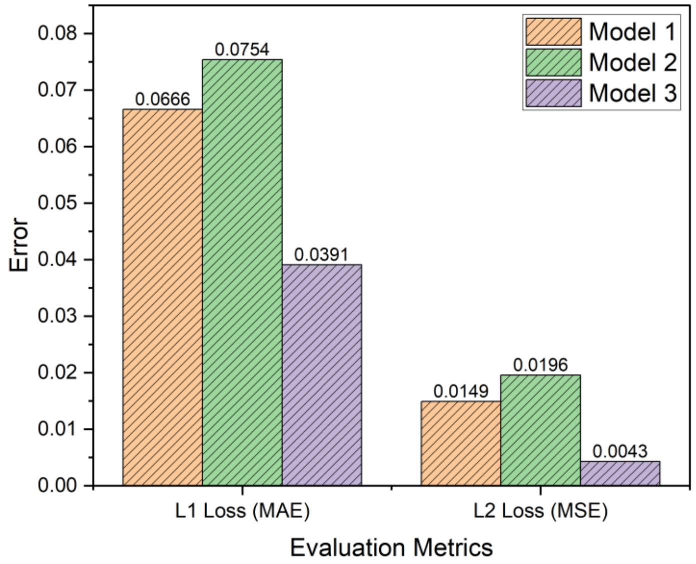
- 32 neurons were selected for first 2 FC-layers
- 256 feature maps used in upsampling

Reduced the parameters from 2.8 million to just 0.7 million

Results and Discussions

Quantitative Comparison

	Phase I			Phase II
Error Metrics	**Model 1**	**Model 2**	**Model 3**	**Modified Model 3**
L2 Loss	0.014857	0.019645	0.004258	**0.002**
PSNR (dB)	19.70	18.09	24.45	**27.8**
Relative Loss	0.101413 (90 % Accuracy)	0.113302 (89 % Accuracy)	0.059131 (95 % Accuracy)	**0.049795**



$$\mathrm{PSNR} = 10\ \log_{10}\left(\frac{\mathrm{L}^2}{\mathrm{MSE}}\right)$$, where L is maximum pixel value

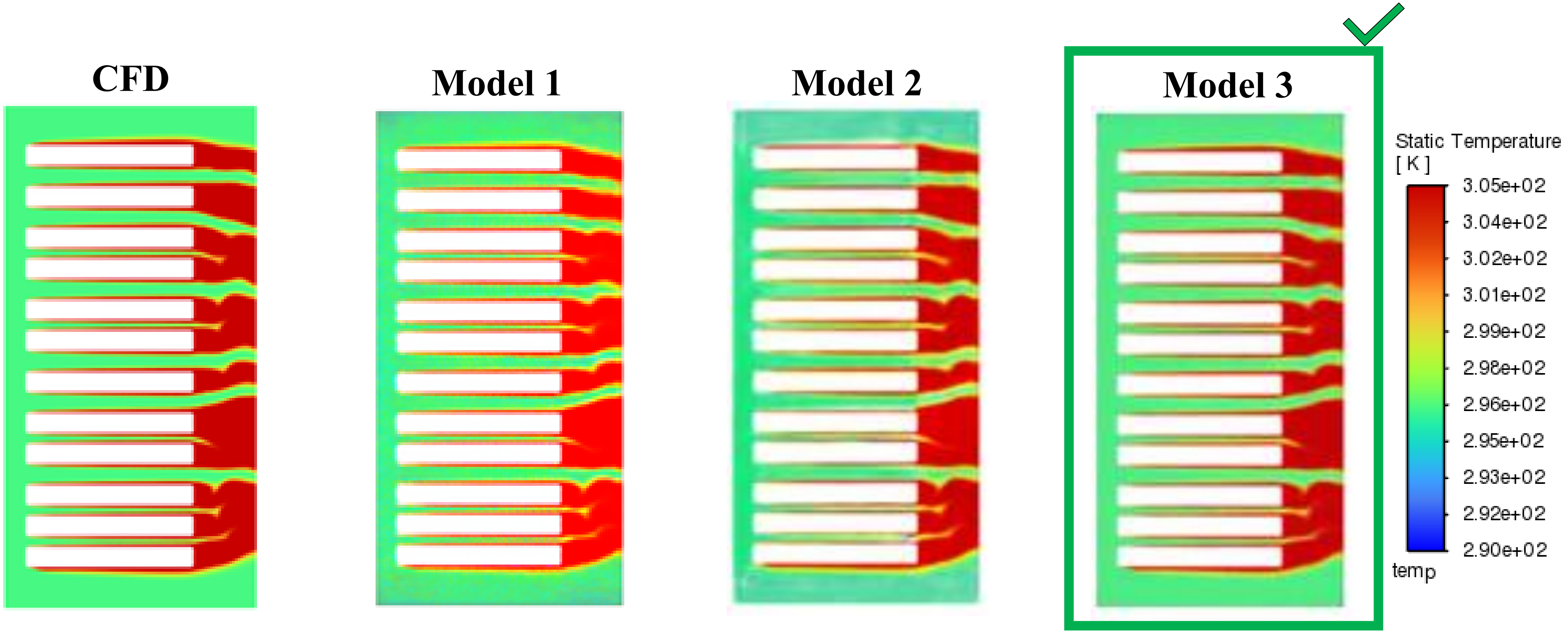
[Inference time]

Results and Discussions

Visual Comparison

Comparison with CFD

Temp = 296 K
Vel = 1.75 m/s
Power = 1601 W

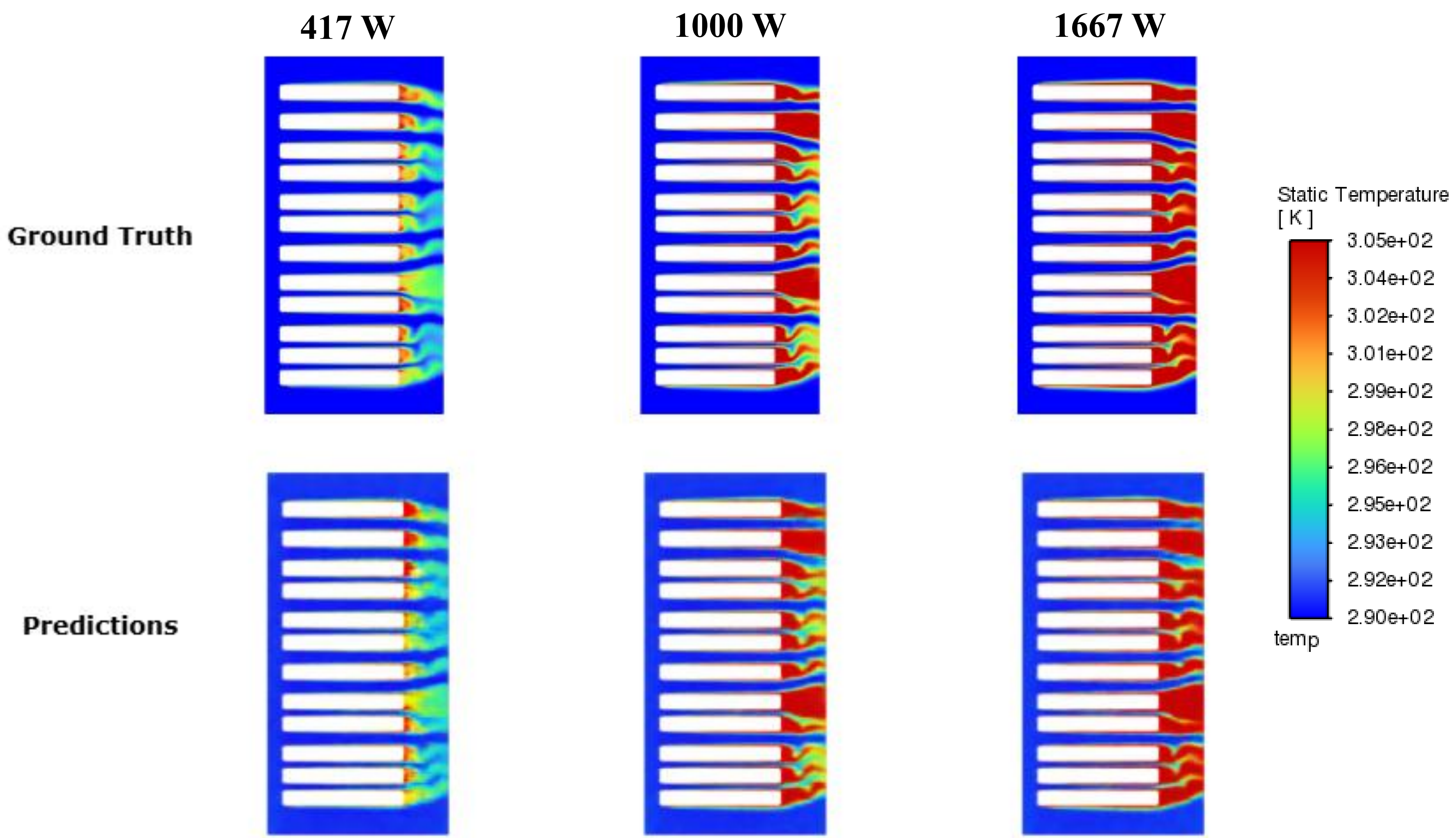


Results and Discussions

Visual Comparison

Increasing Server power

Temp (K) = 290
Vel (m/s) = 1.75

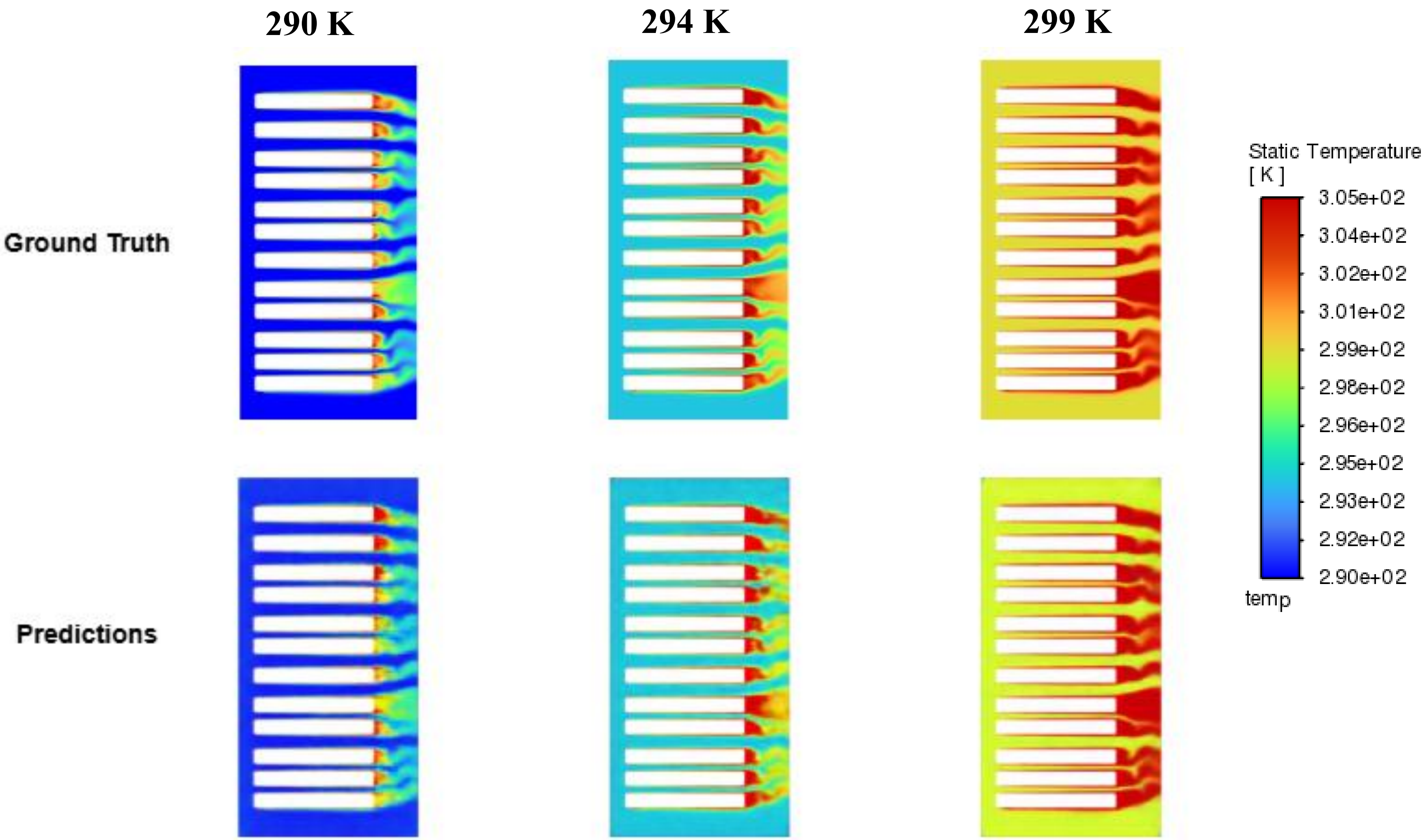


Results and Discussions

Visual Comparison

Increasing Inlet Air Temperature

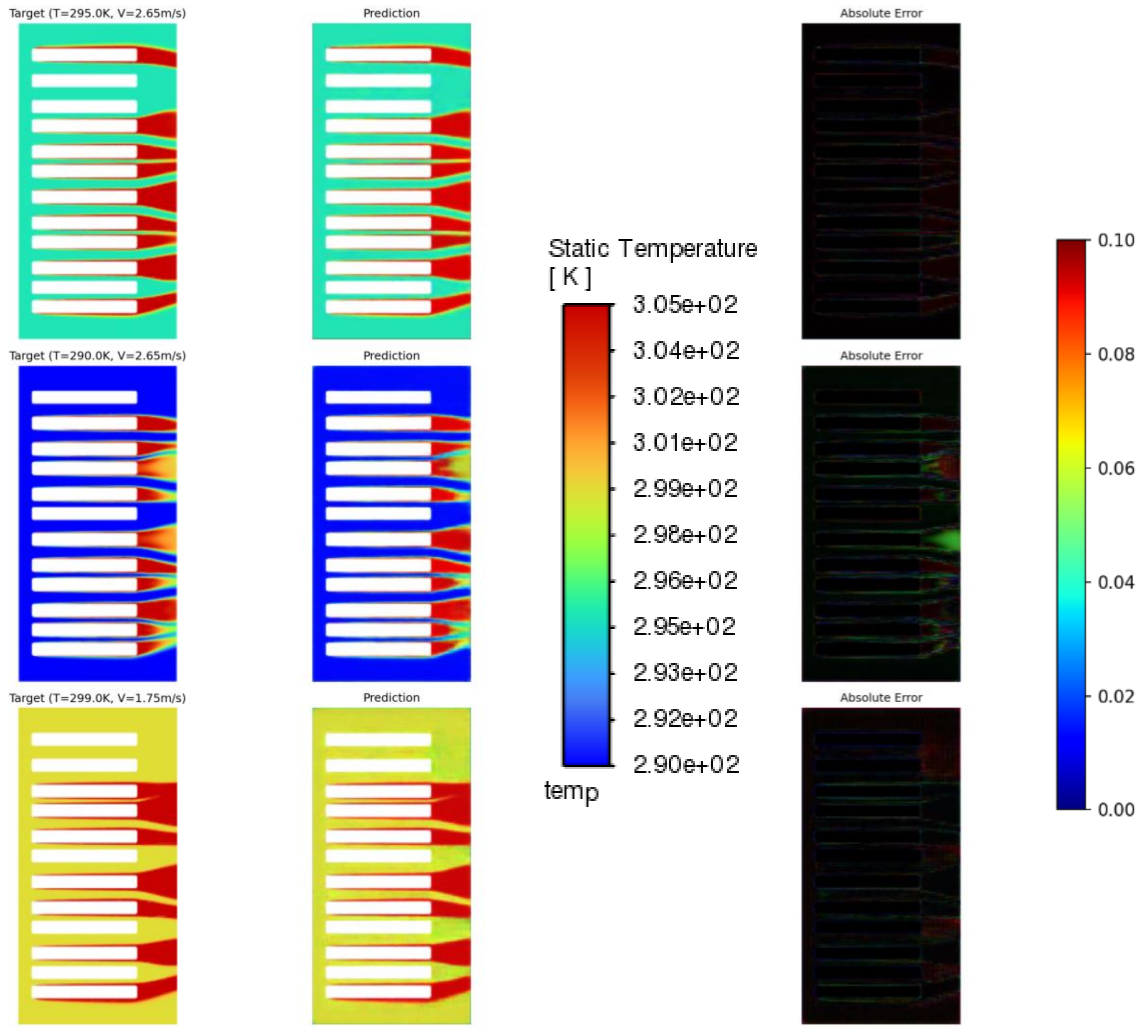
Vel (m/s) = 1.75
Power (W) = 417



Results and Discussions

Visual Comparison

- ❑ Modified Model 3 with 14 inputs
- ❑ Following Representation shows prediction of model for three different scenarios along with Error maps
  - 1^{st} Row: T = 295 K, 2.65 m/s
  - 2^{nd} Row: T = 290 K, 2.65 m/s
  - 3^{rd} Row: T = 299 K, 1.75 m/s



Results and Discussions

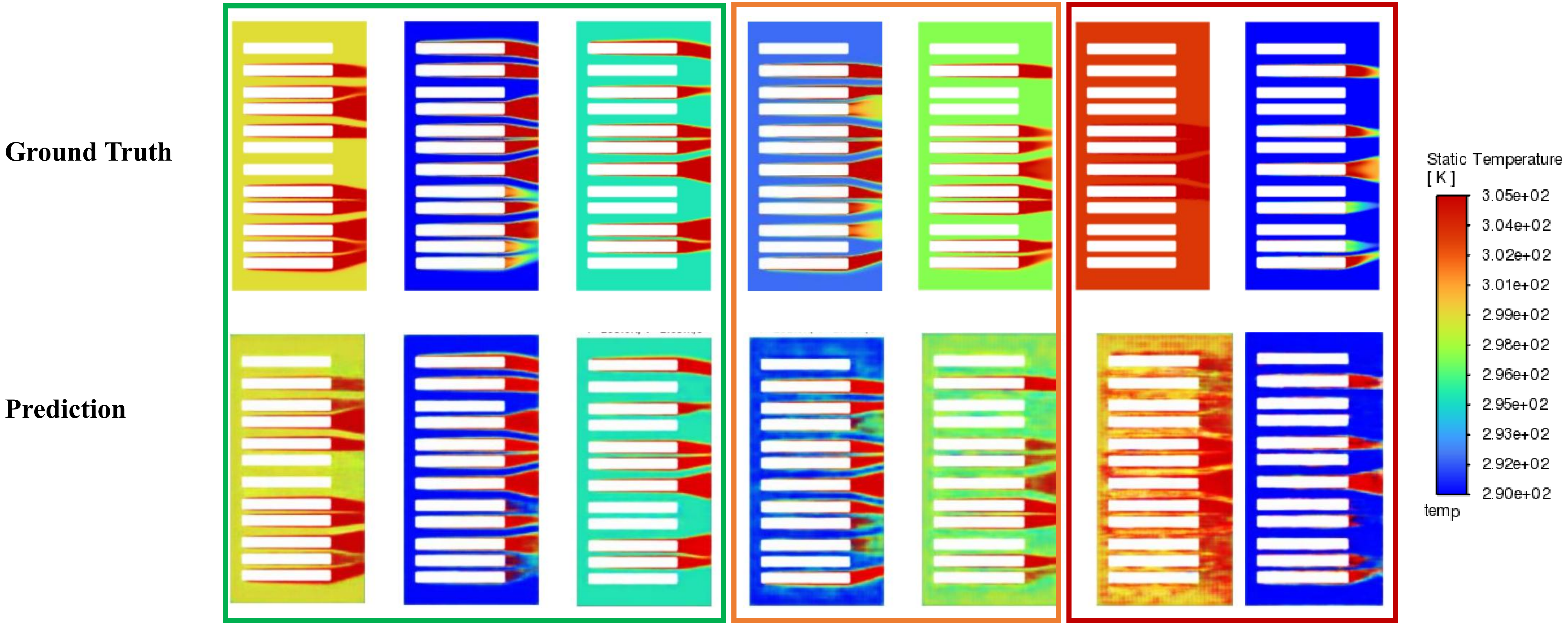
Visual Comparison

- Last of All we assess model prediction for three different scenarios (test set prediction, middle values (interpolation) and extrapolation test

Temperature	Velocity	Server_1 Power	Server_2 Power	Server_3 Power	Server_4 Power	Server_5 Power	Server_6 Power	Server_7 Power	Server_8 Power	Server_9 Power	Server_10 Power	Server_11 Power	Server_12 Power	images
299	2.22	100	25	50	75	100	0	0	75	50	25	25	0	dp1
290	1.75	25	25	50	50	25	75	75	100	75	0	100	100	dp2
295	2.65	0	75	100	0	0	100	100	100	0	50	0	100	dp3
292	1.75	100	0	25	25	50	50	50	75	25	100	100	0	dp4
297	2.22	25	50	0	100	75	25	25	25	0	0	50	0	dp5
303	2.3	0	0	0	0	100	100	100	100	0	0	0	0	dp6
286	1.8	50	25	0	25	0	50	0	50	0	0	50	0	dp7

Results and Discussions

Visual Comparison



Conclusions

- Successfully Developed and evaluated lightweight, decoder-based surrogate models capable of predicting high-resolution temperature distributions from three scalar input parameters.
- Model provide predictions with comparable accuracy to traditional CFD solvers.
- Proposed model, achieves a speed-up of over 6000 times for a 2D case, generating temperature predictions in just 0.07 seconds when compared with CFD simulation.
- The surrogate model demonstrated capability in exploring the design space with millions of combinations.
- This work is a step toward a fast, viable alternative to traditional methods for monitoring and predicting thermal fields in data centers.

Future Directions

- Physics-Informed Machine Learning (PIML): Future efforts will transition toward physics-informed machine learning where the model can penalize when physics laws are violated.
- 3D Modeling and Scalability: The next logical step is to shift toward 3D data and scaling the model to accommodate different geometry layouts.

Reference

- Chen, X., Tu, R., Li, M., Yang, X., & Jia, K. (2023, November). Hot spot temperature prediction and operating parameter estimation of racks in data center using machine learning algorithms based on simulation data. In *Building Simulation* (Vol. 16, No. 11, pp. 2159-2176). Beijing: Tsinghua University Press.

- Zhang, W., Zhang, C., Zhao, Y., Wang, Z., Liu, Y., Zhou, C., & Hu, Y. (2025). Convolutional neural networks-based surrogate model for fast computational fluid dynamics simulations of indoor airflow distribution. *Energy and Buildings*, *326*, 115020.

- Guo, X., Li, W., & Iorio, F. (2016, August). Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 481-490).

- Liu, W., Lian, S., Fang, X., Shang, Z., Wu, H., Zhu, H., & Hu, S. (2023). An open-source and experimentally guided CFD strategy for predicting air distribution in data centers with air-cooling. *Building and Environment*, *242*, 110542.

- Calzolari, G., & Liu, W. (2024, March). Deep learning to develop zero-equation based turbulence model for CFD simulations of the built environment. In *Building Simulation* (Vol. 17, No. 3, pp. 399-414). Beijing: Tsinghua University Press.

- Calzolari, G., & Liu, W. (2023, September). A deep learning accelerator framework for large eddy simulation in the built environment. In *Building Simulation 2023* (Vol. 18, pp. 1264-1271). IBPSA.

- Wang, N., Guo, Y., Huang, C., Tian, B., & Shao, S. (2025). Multi-scale collaborative modeling and deep learning-based thermal prediction for air-cooled data centers: An innovative insight for thermal management. *Applied Energy*, *377*, 124568.

Thankyou ☺