# PointSAGE : Mesh-independent superresolution approach to fluid flow predictions

**Rajat Kumar Sarkar** RAJAT.SARKAR1@TCS.COM
*Researcher*
*TCS Research*

**Krishna Sai Sudhir Aripirala** K.ARIPIRALA@TCS.COM
*Researcher*
*TCS Research*

**Vishal Jadhav** VI.SUJA@TCS.COM
*Scientist*
*TCS Research*

**Sagar Srinivas Sakhinana** SAGAR.SAKHINANA@TCS.COM
*Scientist*
*TCS Research*

**Venkataramana Runkana** VENKAT.RUNKANA@TCS.COM
*Chief Scientist*
*TCS Research*

**Reviewed on OpenReview:** *https://openreview.net/forum?id=MmUfbFD8J0&noteId=54ZvqypOrU*

## Abstract

Computational Fluid Dynamics (CFD) serves as a powerful tool for simulating fluid flow across diverse industries. High-resolution CFD simulations offer valuable insights into fluid behavior and flow patterns, aiding in optimizing design features or enhancing system performance. However, as resolution increases, computational data requirements and time increase proportionately. This presents a persistent challenge in CFD. Recently, efforts have been directed towards accurately predicting fine-mesh simulations using coarse-mesh simulations, with geometry and boundary conditions as input. Drawing inspiration from models designed for super-resolution, deep learning techniques like UNets have been applied to address this challenge. However, these existing methods are limited to structured data and fail if the mesh is unstructured due to its inability to convolute. Additionally, incorporating geometry/mesh information in the training process introduces drawbacks such as increased data requirements, challenges in generalizing to unseen geometries for the same physical phenomena, and issues with robustness to mesh distortions. To address these concerns, we propose a novel framework, **PointSAGE** a mesh-independent network that leverages the unordered, mesh-less nature of Pointcloud to learn the complex fluid flow and directly predict fine simulations, completely neglecting mesh information. Utilizing an adaptable framework, the model accurately predicts the fine data across diverse point cloud sizes, *regardless of the training dataset's dimension.* We have evaluated the effectiveness of

Proceedings of the Data-Centric Machine Learning Workshop(DMLR) at ICLR, 2024 .

PointSAGE on diverse datasets in different scenarios, demonstrating notable results and a significant acceleration in computational time in generating fine simulations compared to standard CFD techniques.

**Keywords:** CFD, Storage Efficiency, Superresolution, Point cloud, PointSAGE

## 1 Introduction

The Navier-Stokes equation stands as a fundamental cornerstone, providing insight into the complex physics that governs scientific and engineering phenomena. However, its intrinsic non-linearity poses challenges. In response, Computational Fluid Dynamics (CFD) has emerged, employing various computational methods to tackle fluid flow complexities. Efforts to predict intricate flows with precision often demand fine resolutions, intensifying computational requirements. In the typical scenario of Direct Numerical Simulation (DNS), fine simulations require a large number of CPU hours and approximately a terabyte (TB) of memory for data storage (Hawkes et al. (2005)). In industrial and real-world contexts, flow dynamics often display significant computational complexity, characterized by turbulence (*pipeline flow*), multi-physics (*aerospace applications*), and multi-phase behaviors (*combustion reactors*). Consequently, simulating these phenomena require extensive computational resources and data storage demands. In contrast, coarse simulations would consume only half the time needed for fine simulations, with memory requirements reduced to about 1/100th for simulating the same physical phenomena.

Hence, coarse grid simulations have become important due to computational efficiency. Yet, the persistent pursuit of understanding complex phenomena drives the ongoing demand for fine-mesh simulations. Recent advancements inspired by super-resolution techniques have introduced deep-learning methodologies for predicting fine-mesh simulations from coarse-mesh counterparts. Utilizing established architectures like MLP (Erichson et al. (2020); Nair and Goza (2020), U-Nets(Sarkar et al. (2023); Pathak et al. (2020)), and GANs (Xie et al. (2018); Bode et al. (2019); Kim et al. (2021); Güemes et al. (2021); Yousif et al. (2021); Bode et al. (2021)), GNNs(Pfaff et al. (2020)) these approaches show promise in overcoming computational challenges to generate fine simulations.

Despite these advancements, current research faces certain challenges. These models adhere to the conventional definition of super-resolution, wherein the fine simulation is *coarsened* with the aid of down-sampling techniques such as max-pooling and nearest neighbor methods (Gao et al. (2021); Bode et al. (2019, 2021); Esmaeilzadeh et al. (2020)). Consequently, the model is trained to learn the mapping between the fine data and its down-sampled counterpart, enabling it to reconstruct the fine data from the down-sampled version. Since this approach retains the physics in the down-sampling process, the training inherently carries a slight bias, resulting in accurate outcomes (Sarkar et al. (2023)). However, real-life scenarios often involve coarse data that is not directly down-sampled from the fine data. As a consequence, these models struggle to predict accurately across different contexts. Recently, researchers have begun utilizing actual coarse data, rather than relying solely on down-sampled fine data, which aligns more closely with real-life scenarios.

However, the current research primarily focuses on structured and regular data, limiting the model's adaptability to various data formats, including unstructured or irregular data (Sarkar et al. (2023); Pathak et al. (2020)). When confronted with unstructured data,

these models fail due to their inability to undergo convolution. Additionally, integrating mesh information during the training phase poses numerous challenges, such as increased data requirements and complexity, leading to extended training time. Training the model on specific geometries restricts its predictive capabilities, especially when assessing unseen or novel geometries for the same physical phenomena, resulting in generalization issues. Moreover, obtaining accurate mesh information in real-world scenarios presents a practical challenge. Consequently, the model's robustness is compromised, affecting its ability to precisely predict fine-mesh simulations in real-world applications.

Consequently, there is a pressing need for a framework that isn't reliant on the geometry or mesh information of the domain. Point clouds offer a comprehensive representation of 3D space by capturing the spatial coordinates of individual points within the domain. In recent times, they have gained prominence due to their ability to handle unstructured data, thanks to their unordered nature (Qi et al. (2017)). This flexibility allows point clouds to capture intricate details and irregularities, making them well-suited for modeling diverse and complex scenarios.

To address these challenges, we introduced **PointSAGE**, a novel *mesh-independent* framework that disregards mesh information by representing data as point clouds sourced from various entities such as cell centers and nodes in the computational domain. This approach affords us the flexibility to manage irregular and unstructured data owing to its unordered nature. PointSAGE incorporates the global feature extraction approach inspired by the "classification network" in the *PointNet* architecture (Qi et al. (2017)), and leverages the potency of *SAGEConv* (Hamilton et al. (2018)) to capture the local inter-dependencies of features within the fluid flow. The model is developed to seamlessly predict fine mesh data, independent of the size/dimensionality of the training sets, *thereby enabling accurate prediction for any value of* $n$. Through comprehensive testing across diverse datasets and scenarios, PointSAGE demonstrated remarkable performance, showcasing significant reductions in training time compared to state-of-the-art (SOTA) techniques. Additionally, our model exhibited enhanced computational efficiency, both in terms of *memory usage and processing time*, when contrasted with conventional CFD methods for fine simulation generation.

## 2 PointSAGE Super-resolution on Point Cloud

In this section, we present the proposed architecture for super-resolving the point cloud of coarse simulation data to match the fine simulation data. The training dataset comprises pairs of coarse mesh data and fine mesh data represented as **point clouds**, denoted as $(C, F)$. The model $f : C \rightarrow F$ is crafted to accurately capture and map the non-linear relationship between the coarse point cloud $C \in \mathbb{R}^{m \times d}$ and the fine point cloud $F \in \mathbb{R}^{n \times d}$, where $m$ and $n$ denote the number of points, and $d$ represents the number of features, with $m \ll n$. A crucial aspect of our approach is our independence from mesh information, solely concentrating on gathering state-variable information at each point cloud. As illustrated in Figure 1, our architecture comprises three components: (a) an Inverse Distance Weighting (IDW) upsampler, (b) a Global Feature Extractor, and (c) a Local Feature Extractor.

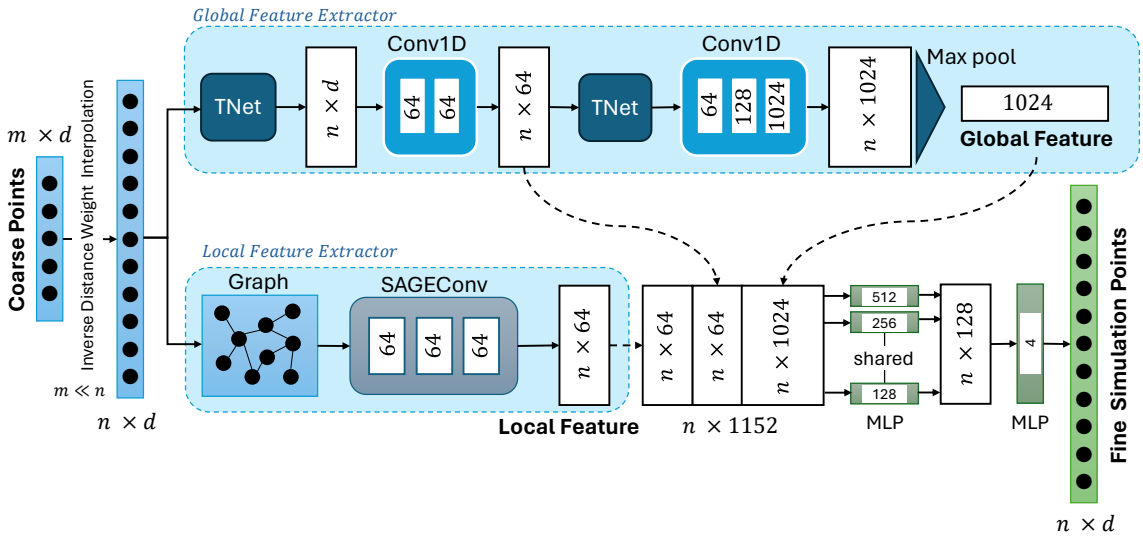



Figure 1: **PointSAGE Architecture** : The coarse data undergoes dimension matching with the fine data through the IDW Up-sampler technique. Subsequently, the up-sampled coarse data is concurrently processed by two modules: the Global (*PointNet*) and Local (*GraphSAGE*) feature extractors. These outputs are then fused through concatenation to accurately predict the fine mesh data.

### 2.1 IDW Upsampler

The coarse point cloud $m$ undergoes an up-sampling process to enhance its resolution, aligning it with the shape of the fine-point cloud $n$. For upsampling, we employ Inverse Distance Weighting with exponential weights. In this interpolated feature $V(n)$ of the coarse mesh at point $n$, we calculate it as:

$$V(n) = \frac{\sum_{i=1}^{n} V_i \cdot e^{-(\frac{D_i}{D_0})^2}}{\sum_{i=1}^{n} e^{-(\frac{D_i}{D_0})^2}}$$

where $D_i$ represents the Euclidean distance between coarse points $m$ and fine points $n$, and $D_0$ is the correlation distance (in this case, the maximum distance is used). After the upsampling, we feed the interpolated point cloud into two other architecture components, as shown in Figure 1.

### 2.2 Global Feature Extractor

The interpolated point cloud $V \in \mathbb{R}^{n \times d}$ undergoes an input transformation using a TNet of dimension $d$. This transformation, achieved through matrix multiplication with TNet output, enables the network to learn robust feature representations invariant to geometric transformations. Subsequently, two Conv1D layers with hidden channels of 64 are employed to extract learned features from the input feature dimension to the hidden dimension. Following this, feature transformation is performed using another TNet of dimension 64 with matrix multiplication, facilitating the capture of complex patterns and structures present

in the data. Multiple Conv1D layers with a max-pooling layer are utilized to aggregate features from all points, extracting a global feature vector of dimension 1024 that represents the entire point cloud. This entire process draws inspiration from the classification section of PointNet.

### 2.3 Local Feature Extractor

Simultaneously, the interpolated point cloud undergoes local feature extractor along with the global feature extractor. In this component, we construct a graph $G = (P, E)$, where $P$ represents the vertices of the graph generated from the $n$ interpolated point cloud, and $E$ denotes the edges connecting the vertices to their neighbors within a fixed radius $r$. The graph construction is achieved using a radius near-neighbor technique with a radius $r = 0.005$, and a maximum of 32 neighbors is given by

$$E = \{(p_i, p_j) \mid \|\mathbf{x}_i - \mathbf{x}_j\|_2 < r\}$$

Subsequently, the constructed graph undergoes multiple SAGEConv layers with a hidden dimension of 64, which aggregate information from neighboring nodes in the graph. This allows for capturing local geometric structures and relationships, facilitating effective local feature learning and extraction of the local feature vector.

The local feature vector is then concatenated with the global feature vector and passed through multiple multi-layer perceptrons (MLPs) to effectively map the coarse point cloud $C$ to the fine point cloud $F$. The Mean Squared Error (MSE) is employed as the loss function for training the model. Importantly, our methodology deliberately avoids incorporating any mesh information throughout this entire process. This ensures that our model remains entirely mesh-independent, contributing to its robustness and versatility in handling diverse mesh representations such as irregularly structured and unstructured grids.

## 3 Results and Discussion

To demonstrate the effectiveness of our model, we conducted experiments on three diverse datasets: a forward-facing step, a lid-driven cavity, and methane combustion simulations. We evaluated our model's performance using 2D point cloud data from the forward-facing step simulation and extended it to 3D lid-driven cavity simulation data. Additionally, we compared our model to recent work, specifically PIUNet, Sarkar et al. (2023), which operates on regular grid fluid flow prediction, using 2D methane combustion data. This comparison aims to highlight the advantages of our approach, PointSAGE, utilizing point cloud data from the simulation.

**Forward-Step:** The transient CFD simulation of the forward step provides valuable insights into the complex flow phenomena associated with separated and reattached flows in a step-like configuration. We specifically focused on predicting the shock wave generated by supersonic flow at the inlet to a rectangular geometry with a step near the inlet region. Details of the CFD simulation are explained in the appendix A.1. To showcase our approach's effectiveness, we demonstrated the model's performance in two scenarios. In the first scenario, we varied the inlet velocity ($U_\infty \in [2, 5]$ m/s) for a given Aspect Ratio (AR) of 3. In the second scenario, we varied both the inlet velocity ($U_\infty \in [2, 5]$ m/s) and

the Aspect Ratio ($AR \in [3, 6]$). Table 1 shows that PointSAGE outperforms other mesh-independent approaches in both scenarios. The problem involves four features: $x$-direction velocity ($U_x$), $y$-direction velocity ($U_y$), pressure ($Pa$), and Mach Number ($Ma$). In terms of RMSE, we achieved approximately 70% and 20% enhancement in all features compared to the second-best results in scenarios 1 and 2, respectively.

Table 1: Performance of PointSAGE fine mesh prediction on 2D forward step dataset

| **Scenario 1:** Varying inlet velocity for an Aspect Ratio (AR) of 3 (with a 2.4m block length beyond the step)(3352 pts → 42702 pts) | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Algorithm** | **MAE** | **RMSE** | **$R^2$** | | **MAE** | **RMSE** | **$R^2$** | | **MAE** | **RMSE** | **$R^2$** | | **MAE** | **RMSE** | **$R^2$** |
| $U_x$ | SAGEConv | 0.9025 | 1.0919 | 0.1695 | $U_y$ | 0.2434 | 0.3847 | 0.5572 | Pressure | 2.4264 | 4.5794 | 0.1626 | Mach No. | 0.5665 | 0.7725 | 0.6726 |
| | PointNET | 0.6428 | 1.017 | 0.3126 | | 0.2687 | 0.4065 | 0.4973 | | 2.0413 | 4.0606 | 0.3416 | | 0.5101 | 0.758 | 0.6784 |
| | **PointSAGE** | **0.1651** | **0.3418** | **0.8733** | | **0.0941** | **0.1668** | **0.8996** | | **0.7323** | **1.6079** | **0.8691** | | **0.1476** | **0.275** | **0.9415** |
| **Scenario 2:** Training on AR 3 and 4, validating on AR 5, testing on AR 6 (6082 pts → 91302 pts), with varying inlet velocity and aspect ratio. | | | | | | | | | | | | | | | |
| | **Algorithm** | **MAE** | **RMSE** | **$R^2$** | | **MAE** | **RMSE** | **$R^2$** | | **MAE** | **RMSE** | **$R^2$** | | **MAE** | **RMSE** | **$R^2$** |
| $U_x$ | SAGEConv | 0.6254 | 0.8411 | -0.9743 | $U_y$ | 0.2324 | 0.3138 | -0.4679 | Pressure | 1.6785 | 2.6554 | -0.3859 | Mach No. | 0.667 | 0.8497 | -0.8944 |
| | PointNET | 0.5009 | 0.7747 | -0.2627 | | 0.2052 | 0.2985 | -0.1746 | | 1.3818 | 2.2660 | 0.0150 | | 0.5147 | 0.7196 | -0.0550 |
| | **PointSAGE** | **0.2810** | **0.5008** | **0.3125** | | **0.2007** | **0.2859** | **-0.0962** | | **1.1817** | **2.0587** | **0.0380** | | **0.2854** | **0.4610** | **0.4408** |

From Figure 2, it is evident that at time step $t = 3.5$ s, there is a noticeable generation of a shock wave near the forward step due to an inlet velocity of 4.465 m/s (equivalent to Mach 4 in this fluid medium where the velocity of sound is $a = \sqrt{\gamma RT} = 1$ m/s). Subsequently, after traveling a certain distance, the shock wave gets reflected from the upper surface, creating a reflected shock. Our model, PointSAGE, successfully captures all these phenomena with a maximum absolute pressure error $E$ of 10 in the reflected shock region, where $E = |\hat{X} - X|$, and $\hat{X}$ refers to the prediction while $X$ refers to the fine mesh data.

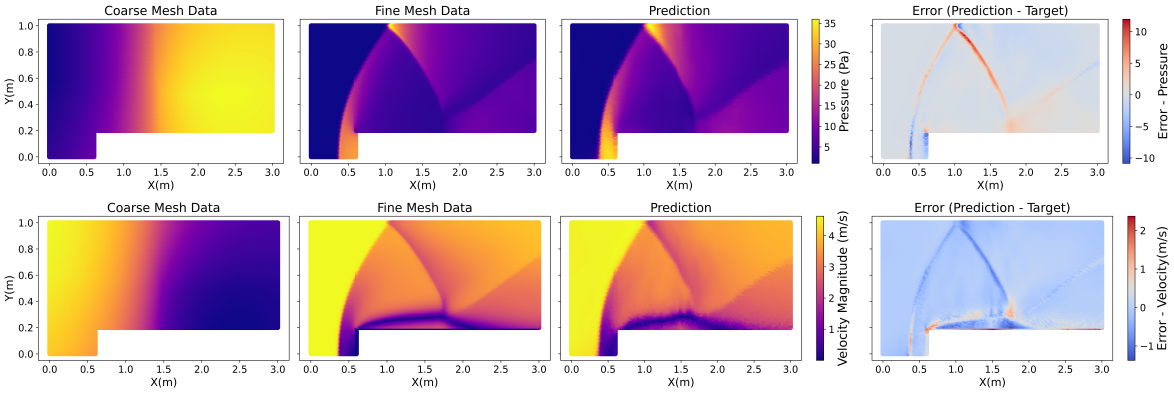



Figure 2: Scenario 1: Pressure and Velocity prediction at t= 3.5s for inlet velocity 4.465 m/s for an AR 3 dataset.

**Lid-Driven Cavity:** Hanna et al. (2017) This case study on the lid-driven cavity aims to showcase our model's predictive capabilities. It demonstrates two key aspects: Firstly, the model's ability to learn and predict the turbulence aspect of the flow, notably the bottom-right vortex in the cavity, as depicted in Figure 3, where the velocity contour is plotted. Secondly, its effectiveness in handling unseen geometries or conditions after training on various scenarios, as evidenced by Table 2. Our model achieves comparable accuracy

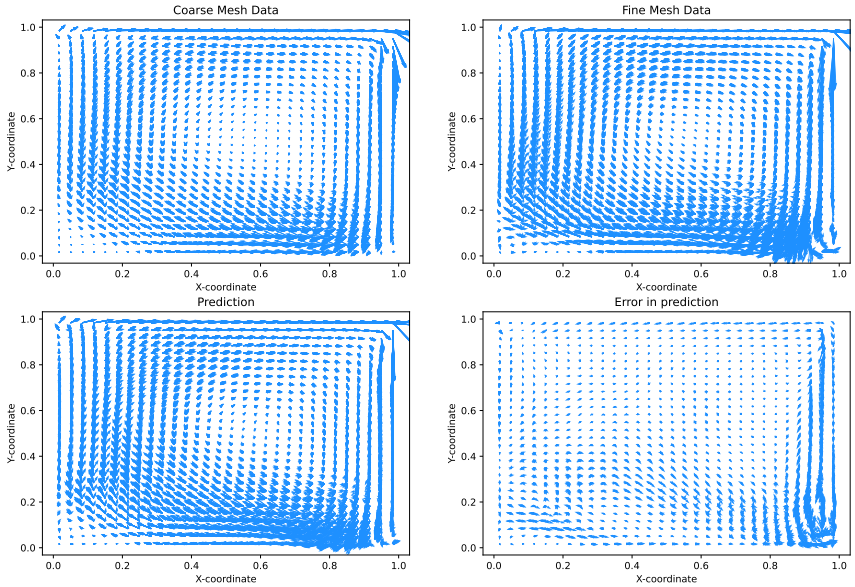



Figure 3: Scenario 1: Velocity prediction for the case of Reynolds number interpolation

Table 2: Performance of PointSAGE fine mesh prediction on 3D Lid Driven cavity dataset (where Time is Training time)

| | Features | Algorithms | MSE ($1e^{-4}$) | $R^2$ | Time (sec) | | MSE ($1e^{-4}$) | $R^2$ | Time (sec) | | MSE ($1e^{-4}$) | $R^2$ | Time (sec) | | MSE ($1e^{-4}$) | $R^2$ | Time (sec) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Re Interpolation | $U_x$ | CG-CFD | 1 | 0.915 | 660 | Re Extrapolation | 1 | - | 660 | Re & GS Interpolation | 1 | - | 660 | Re & GS Extrapolation | 1 | - | 780 |
| | $U_x$,$U_y$,$U_z$ | UNet | 1.3 | 0.971 | 600 | | 1.4 | 0.944 | 660 | | - | - | - | | - | - | - |
| | | SAGEConv | 2.67 | 0.965 | 80 | | 3.6 | 0.933 | 83 | | 3 | 0.959 | 93 | | 1.7 | 0.978 | 41 |
| | | PointNET | 2.7 | 0.967 | 26 | | 3.3 | 0.937 | 29 | | 3.1 | 0.961 | 28 | | 1.7 | 0.98 | 13 |
| | | **PointSAGE** | 2.6 | 0.959 | 120 | | 3.5 | 0.924 | 120 | | 3 | 0.952 | 156 | | 2.3 | 0.968 | 50 |

with existing benchmark techniques in scenarios such as Re interpolation or Re&GS extrapolation, all while requiring significantly less training time— notably, five times faster. In addition to our developed model, we have also utilized other methods such as SAGEConv and PointNet to demonstrate the effectiveness of these mesh-independent approaches in learning and predicting fine-mesh simulations with notable accuracy. While these methods excel in simpler datasets like the one presented here, our model excels in managing more intricate data scenarios.

**Methane Combustion:** Yang et al. (2019) This case study on methane combustion aims to highlight the contrast between two methodologies: PointSAGE and recent work as mentioned above, PIUNet, which relies solely on a regular mesh grid. We assess various outputs, including the adiabatic flame temperature ($T_{adia}$), $x$-direction velocity ($U_x$), $y$-direction velocity ($U_y$), and mass fractions of species $CH_4$, $O_2$, and $CO_2$. Analyzing the data from Table 3, we observe that PointSAGE achieves comparable results to PIUNet in

terms of adiabatic temperature. However, our model outperforms or ranks second best in other features compared to benchmark algorithms. These findings demonstrate that not only does PointSAGE offer a mesh-independent approach, facilitating super-resolution on any mesh or geometry, but it also predicts finer mesh simulation results with comparable accuracy.

Table 3: Performance of PointSAGE fine mesh prediction (1000 pts → 50000 pts) on 2D Methane Combustion dataset

| | | Algorithm | MAE | RMSE | $R^2$ | | MAE | RMSE | $R^2$ | | MAE | RMSE | $R^2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Fluid properties** | Temperature | UNet | 13.224 | 30.718 | 0.9963 | $U_x$ | 0.0177 | 0.0296 | 0.9839 | $U_y$ | 0.0152 | 0.0324 | 0.9830 |
| | | PIUNet | **10.385** | **20.954** | **0.9984** | | 0.0164 | 0.0286 | **0.9862** | | 0.0158 | 0.0324 | 0.9835 |
| | | SAGEConv | 18.209 | 30.075 | 0.9959 | | 0.0079 | 0.0107 | 0.9924 | | 0.0109 | 0.0153 | **0.9940** |
| | | PointNET | 27.543 | 46.745 | 0.9847 | | 0.0075 | 0.0105 | 0.9836 | | 0.0154 | 0.0222 | 0.9735 |
| | | **PointSAGE** | 17.296 | 28.701 | 0.9947 | | **0.0071** | **0.0105** | 0.9855 | | **0.0104** | **0.0151** | 0.9930 |
| **Mass Fraction** | $CH_4$ | UNet | 0.0140 | 0.0140 | 0.9938 | $O_2$ | 0.0059 | 0.0116 | 0.9870 | $CO_2$ | 0.0058 | 0.0094 | 0.9564 |
| | | PIUNet | 0.0138 | 0.0138 | 0.9954 | | 0.0030 | 0.0106 | 0.9888 | | 0.0018 | 0.0051 | **0.9844** |
| | | SAGEConv | **0.0089** | **0.0128** | **0.9987** | | 0.0024 | 0.0041 | 0.9974 | | 0.0013 | 0.0024 | 0.9639 |
| | | PointNET | 0.0125 | 0.0191 | 0.9932 | | 0.0028 | 0.0053 | 0.9954 | | 0.0016 | 0.0033 | -386.93 |
| | | **PointSAGE** | 0.0100 | 0.0140 | 0.9971 | | **0.0024** | **0.0038** | **0.9975** | | **0.0012** | **0.0022** | 0.9344 |

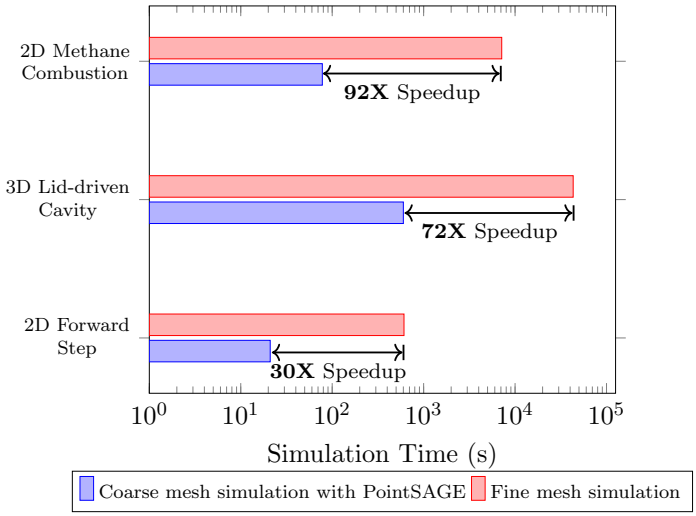



Figure 4: Comparison of Speedup Achieved by PointSAGE in Accelerated CFD Simulations: The *blue bars* represent the time taken for coarse mesh simulation along with the inference time of PointSAGE for predicting fine mesh simulation, while the *red bars* represent the simulation time for fine mesh simulation using the CFD solver OpenFOAM.

From Figure 4, it is evident that incorporating PointSAGE in the simulation process significantly reduces computation time compared to traditional CFD simulations. Specifically, simulations with PointSAGE achieve notable speedups, with a *30X, 72X, and 92X* improvement in simulation time for the 2D Forward Step, 3D Lid-driven Cavity, and 2D Methane Combustion cases, respectively. These speedups are crucial for real-world applications where computational efficiency is paramount. Furthermore, PointSAGE enables

accurate predictions even in unseen scenarios, as demonstrated in previous results. This dual capability positions our model as a powerful tool for accelerating computational fluid dynamics simulations while maintaining high prediction accuracy across various domains and scenarios. Due to this significant acceleration in computational speed and reliable accuracy, we can now rely on coarse simulations, thus enhancing *storage and memory efficiency* in CFD simulations. Nevertheless, the model has certain *limitations*. The size of the 3D point cloud we processed remains relatively modest. As this dimension expands, the "GraphSAGE" component within our model will lead to extended training periods due to the substantial volume of message-passing, thereby increasing the model's complexity. Consequently, in our future works, we aim to employ advanced versions of GNNs to tackle this challenge and guarantee the model's adaptability to larger point cloud dimensions. The framework operates under supervised learning, learning from observed coarse and fine mesh data to predict fine mesh data from unseen coarse mesh data. Future work aims to develop an unsupervised learning model for broader applicability.

## 4 Conclusion

We introduce PointSAGE, a model for superresolution using point clouds, showcasing its ability to predict fine-mesh data solely from coarse-mesh data without prior knowledge of mesh characteristics. Leveraging point cloud data, our model demonstrates robust performance across diverse datasets and unseen geometries, such as different aspect ratios and varying inlet conditions, highlighting its generalizability. The framework's adaptability enables it to predict fine mesh data of any shape/size, regardless of the dimension of the training data. In a case study of forward-facing step simulation, PointSAGE accurately captures primary shock formation and reflected shocks, achieving substantial enhancements in RMSE and MAE compared to existing deep learning techniques. Similarly, in Lid-driven cavity simulations, our model exhibits superior predictive capability in turbulent scenarios within a 3D computational domain through various scenarios, including Reynolds number extrapolation and Grid Size interpolation, with significant reductions in training time and improved MSE compared to benchmarks. The key innovation lies in our method, which eliminates the need for detailed mesh information, showcasing impressive results and setting the stage for future developments in the field. By adopting a mesh-independent approach, our work aims to revolutionize the prediction of fine-mesh simulations for fluid flows, offering a more versatile and efficient solution in CFD applications. However, the model faces limitations, particularly with the relatively small size of the 3D point cloud it handles. As the size increases, the GraphSAGE component results in longer training times due to extensive message-passing, increasing model complexity. Therefore, in future work, we plan to utilize advanced GNN versions to address this scalability issue. Also, the framework currently uses supervised learning with observed data to predict unseen fine mesh data. Future plans involve exploring unsupervised learning for broader applicability.

## Impact Statement

Our novel framework, PointSAGE, represents a significant advancement in the field of Computational Fluid Dynamics (CFD) by addressing key challenges associated with fine-

mesh simulations. By leveraging the unordered, mesh-independent nature of point clouds, PointSAGE eliminates the need for intricate mesh information, making it adaptable to diverse and irregular data formats. Our model incorporates global feature extraction and local inter-dependency capture techniques, resulting in accurate predictions across a *wide range of point cloud sizes, irrespective of the dimensions of the training datasets*. This capability ensures robust performance and generalization to unseen geometries, enhancing the model's utility in real-world applications. The impact of our work is multifaceted. Firstly, PointSAGE offers a *significant acceleration in computational time and reduction in memory usage* for generating fine simulations compared to traditional CFD techniques, making it an invaluable tool engineers and scientists working in fluid dynamics. Secondly, by eliminating the reliance on mesh information, our model reduces the complexity associated with data preparation and training, thereby streamlining the workflow and reducing computational resources. Lastly, the ability of PointSAGE to accurately predict fine data across diverse point cloud sizes enhances its versatility and applicability to a wide range of scenarios, from aerodynamics to environmental fluid dynamics. Overall, PointSAGE represents a paradigm shift in CFD simulations, offering a flexible, efficient, and accurate solution to the challenges posed by fine-mesh simulations. Its impact extends beyond the realm of fluid dynamics, with potential applications in various fields requiring predictive modeling of complex systems.

## References

M. Bode, M. Gauding, K. Kleinheinz, and H. Pitsch. Deep learning at scale for subgrid modeling in turbulent flows: regression and reconstruction. In *International Conference on High-Performance Computing*, pages 541–560. Springer, 2019.

M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, and H. Pitsch. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. *Proceedings of the Combustion Institute*, 38(2):2617–2625, 2021.

N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz. Shallow neural networks for fluid flow reconstruction with limited sensors. *Proceedings of the Royal Society A*, 476(2238):20200097, 2020.

S. Esmaeilzadeh, K. Azizzadenesheli, K. Kashinath, M. Mustafa, H. A. Tchelepi, P. Marcus, M. Prabhat, A. Anandkumar, et al. Meshfreeflownet: A physics-constrained deep continuous space-time super-resolution framework. In *SC20: International Conference for High-Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE, 2020.

H. Gao, L. Sun, and J.-X. Wang. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, 33(7), 2021.

A. Güemes, S. Discetti, A. Ianiro, B. Sirmacek, H. Azizpour, and R. Vinuesa. From coarse wall measurements to turbulent velocity fields through deep learning. *Physics of Fluids*, 33(7), 2021.

W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs, 2018.

B. N. Hanna, N. T. Dinh, R. W. Youngblood, and I. A. Bolotnov. Coarse-grid computational fluid dynamic (cg-cfd) error prediction using machine learning. *arXiv preprint arXiv:1710.09105*, 2017.

E. R. Hawkes, R. Sankaran, J. C. Sutherland, and J. H. Chen. Direct numerical simulation of turbulent combustion: fundamental insights towards predictive models. In *Journal of Physics: Conference Series*, volume 16, page 65. IOP Publishing, 2005.

H. Kim, J. Kim, S. Won, and C. Lee. Unsupervised deep learning for super-resolution reconstruction of turbulence. *Journal of Fluid Mechanics*, 910:A29, 2021.

N. J. Nair and A. Goza. Leveraging reduced-order models for state estimation using deep learning. *Journal of Fluid Mechanics*, 897:R1, 2020.

J. Pathak, M. Mustafa, K. Kashinath, E. Motheau, T. Kurth, and M. Day. Using machine learning to augment coarse-grid computational fluid dynamics simulations. *arXiv preprint arXiv:2010.00072*, 2020.

T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2017.

R. K. Sarkar, R. Majumdar, V. Jadhav, S. S. Sakhinana, and V. Runkana. Redefining super-resolution: Fine-mesh pde predictions without classical simulations. *arXiv preprint arXiv:2311.09740*, 2023.

Y. Xie, E. Franz, M. Chu, and N. Thuerey. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018.

Q. Yang, P. Zhao, and H. Ge. reactingfoam-sci: An open source cfd platform for reacting flow simulation. *Computers & Fluids*, 190:114–127, 2019.

M. Z. Yousif, L. Yu, and H.-C. Lim. High-fidelity reconstruction of turbulent flow from spatially limited data using enhanced super-resolution generative adversarial network. *Physics of Fluids*, 33(12), 2021.

# Appendix A. Appendix

## A.1 Case Study 1: Forward Facing Step Simulation

In this case study we investigate transient simulation of supersonic flow over a forward-facing step using sonicFoam solver in OpenFOAM. The sonicFoam solver is designed to solve compressible trans-sonic/supersonic laminar gas flow. The problem description involves a flow of Mach 3 at an inlet to a rectangular geometry with a step near the inlet region that generates shock waves and propagates downstream and get reflected from the walls and creates reflected shocks in the remaining length after the forward step till the time it reaches its steady state. This case study we have selected from the OpenFOAM tutorial this link.

### A.1.1 Problem description

**Solution domain**

The 2D computational domain features a step with a height of 20% located at a distance of 0.6m from the inlet as shown in the Figure 5. The experiment is conducted in a gas medium with a speed of sound given by $\sqrt{\gamma RT} = 1$ m/s. Thus, at the inlet, the flow is supersonic with a Mach number of 3 ($U_\infty = 3$ m/s), along with a pressure of 1 Pa and a temperature of 1 K. The aspect ratio of the defined geometry in this case study is expressed as the ratio

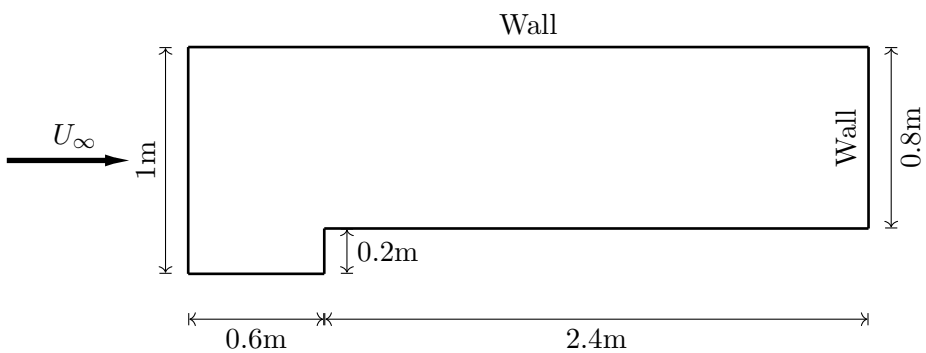



Figure 5: Computational Domain of a 2D Forward Facing Step Simulation

of the length of the rectangular domain (3m) to its height (1m), as shown below:

$$\text{Aspect ratio} = \frac{\text{Length}}{\text{Height}} = \frac{3\,\text{m}}{1\,\text{m}}$$

**Governing equations**

Mass continuity:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \tag{1}$$

Ideal gas:

$$p = \rho RT \tag{2}$$

Momentum equation for Newtonian fluid:

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) - \nabla \cdot \mu \nabla \mathbf{U} = -\nabla p \tag{3}$$

The energy equation for fluid (ignoring some viscous terms):

$$\frac{\partial(\rho e)}{\partial t} + \nabla \cdot (\rho \mathbf{U} e) - \nabla \cdot \left( \frac{k}{C_v} \nabla e \right) = p \nabla \cdot \mathbf{U} \tag{4}$$

**Initial Conditions:**

$$U = 0\,\mathrm{m/s}, \quad p = 1\,\mathrm{Pa}, \quad T = 1\,\mathrm{K}$$

**Boundary Conditions:**

- **Inlet (left):**

  FixedValue for velocity: $U = 3\,\mathrm{m/s}$ (Mach 3)
  Pressure: $p = 1\,\mathrm{Pa}$
  Temperature: $T = 1\,\mathrm{K}$

- **Outlet (right):**

  ZeroGradient on $U, p,$ and $T$

- **No-slip adiabatic wall (bottom)**

- **Symmetry plane (top)**

**Transport Properties:**

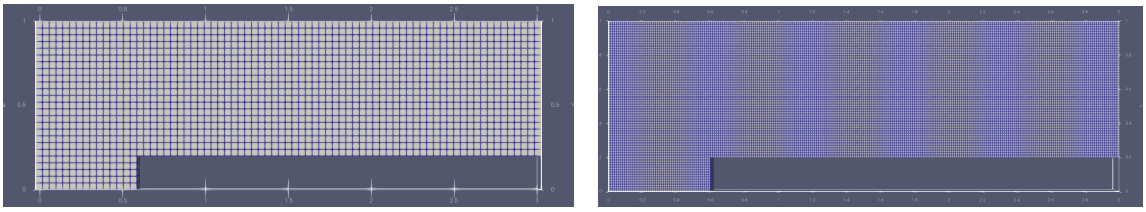
*Laminar* Dynamic viscosity of air: $\mu = 18.1\,\mu\mathrm{Pas}$

**Thermodynamic Properties:**

Specific heat at constant volume: $C_v = 1.78571\,\mathrm{J/kgK}$
Gas constant: $R = 0.714286\,\mathrm{J/kgK}$
Conductivity: $k = 32.3\,\mu\mathrm{W/mK}$

### A.1.2 Mesh Description

The mesh is generated using the *blockMesh* utility, dividing the domain into uniform rectangular cells. For the fine mesh, the cells have dimensions of 0.03 m in the $x$-direction and 0.025 m in the $y$-direction, resulting in 42702 points for the point cloud. Conversely, the coarse mesh divides the domain into cells with dimensions of 0.12 m in the $x$-direction and 0.1 m in the $y$-direction, yielding 3352 points for the coarse point cloud used in our neural network.



(a) Coarse Mesh (b) Fine Mesh

Figure 6: Mesh Description: Aspect Ratio 3

### A.1.3 Scenario 1

In this scenario, we conducted experiments using a dataset where the inlet velocity was varied in the range of 2 m/s to 5 m/s, with intervals of 0.25 m/s, while maintaining a constant aspect ratio of 3, as mentioned earlier. The simulations were performed on an Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz. For deep learning experiments, we partitioned the dataset into 80%/10%/10% for training, validation, and testing respectively, and executed the entire experiment on a Tesla P100 GP with 16GB VRAM.

**Results**

The simulations conducted were of a transient nature, and PointSAGE demonstrated commendable accuracy in predicting features such as pressure and velocity at different time intervals, as illustrated in Figure 8 and Figure 9 respectively. Both figures reveal that our PointSAGE model effectively captures the propagation of shocks and their reflection within the rectangular domain following the step location. The training and validation for the PointSAGE training can be observe in this Figure 7.

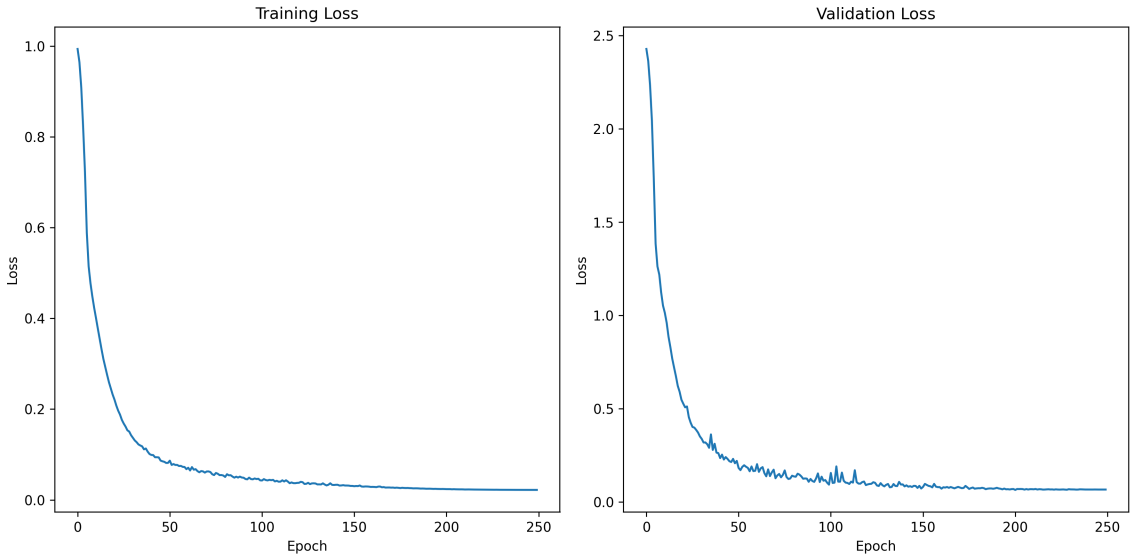
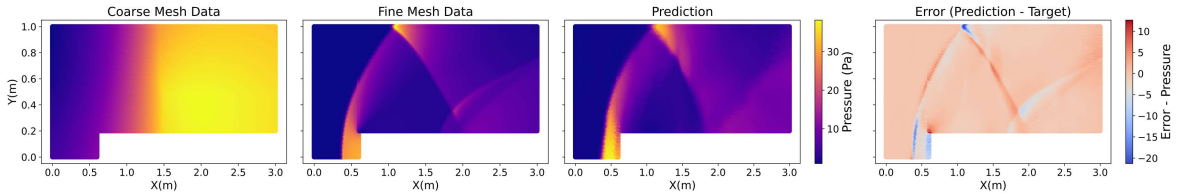
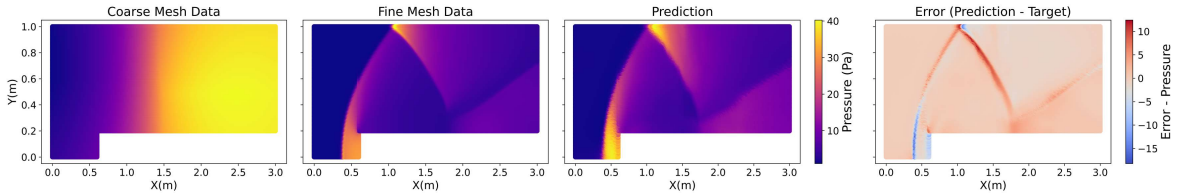


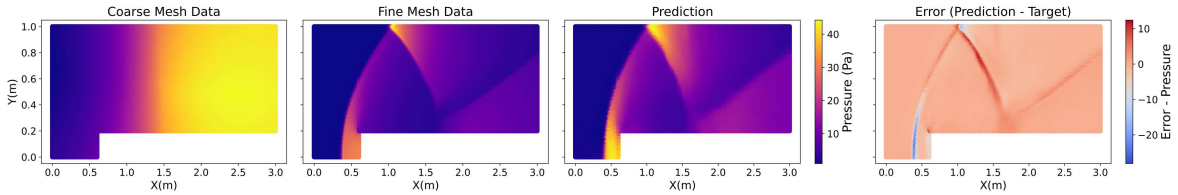

Figure 7: PointSAGE Training and Validation Loss



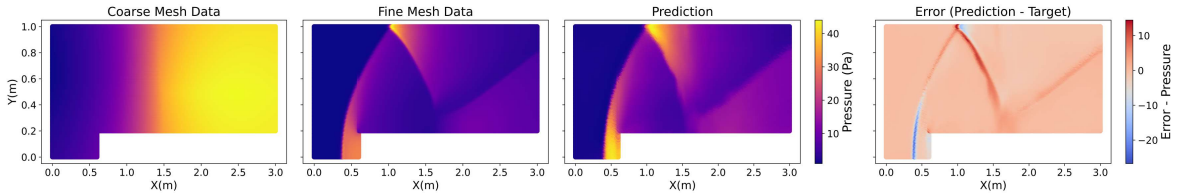

(a) time-step $t = 2s$




(b) time-step $t = 4.5s$

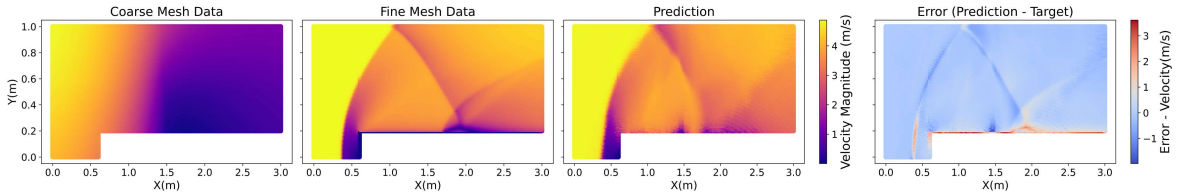



(c) time-step $t = 7s$

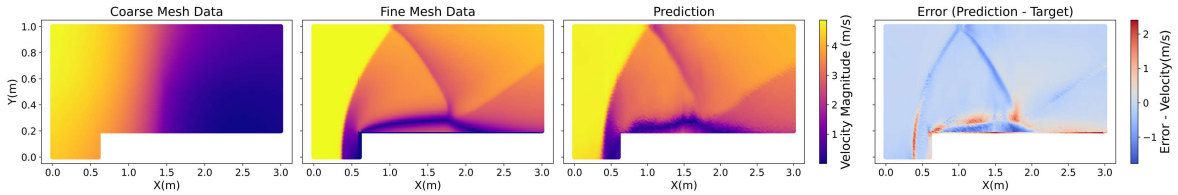



(d) time-step $t = 9.5s$

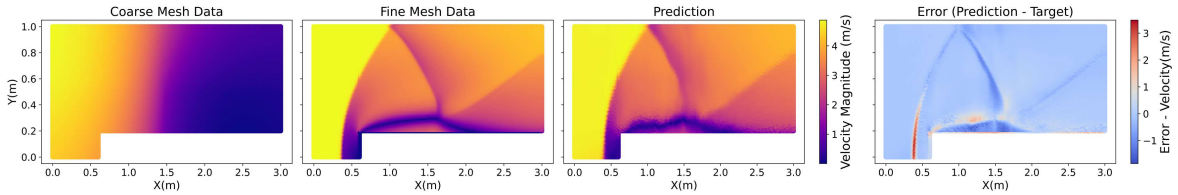
Figure 8: Pressure distribution from PointSAGE-predicted fine mesh simulation at various time steps, corresponding to an inlet velocity $U_\infty$ of 4.875 m/s.



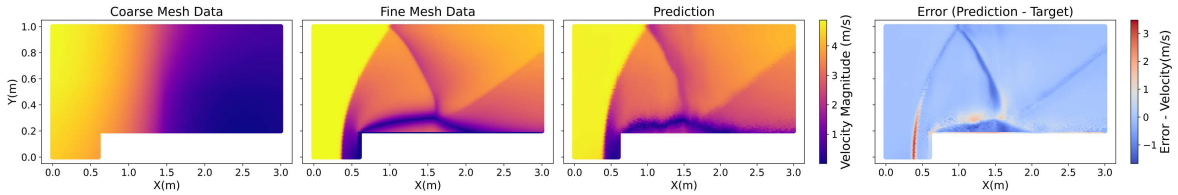

(a) time-step $t = 2s$




(b) time-step $t = 4.5s$




(c) time-step $t = 7s$




(d) time-step $t = 9.5s$

Figure 9: Velocity distribution from PointSAGE-predicted fine mesh simulation at various time steps, corresponding to an inlet velocity $U_\infty$ of 4.875 m/s.

### A.1.4 Scenario 2:

In this particular scenario, we conducted experiments utilizing a dataset wherein the inlet velocity was systematically varied within the range of 2 m/s to 5 m/s, with intervals of 0.5 m/s. Additionally, we varied the aspect ratio within the range of 3 to 6. The variation in aspect ratio involves an increase in the length after the step location at 0.6 m. For instance, in the case of an aspect ratio of 3, the length of the section after the step is 2.4 m (total length = 0.6 + 2.4 = 3m). On the other hand, for an aspect ratio of 4, the length of the section after the step is 3.4 m (total length = 0.6 + 3.4 = 4m). In the context of deep learning experiments, we partitioned the dataset for training using aspect ratios 3 and 4, for validation with aspect ratio 5, and for testing with aspect ratio 6.

**Results**

The objective of this scenario is to evaluate the model's proficiency in effectively understanding and adapting to physical phenomena, specifically shock formation and reflection, within a given aspect ratio. Furthermore, the model is challenged to extend its predictions to another aspect ratio, adding complexity as an increase in the length after the step leads to intensified shock reflection and sudden alterations in flow behavior downstream. As illustrated in Figure 10, PointSAGE demonstrates satisfactory predictions for essential features such as pressure and velocity. This success underscores the model's ability to effectively capture and forecast the dynamic behaviors of shocks under varying aspect ratios, emphasizing its efficacy in handling complex flow phenomena.

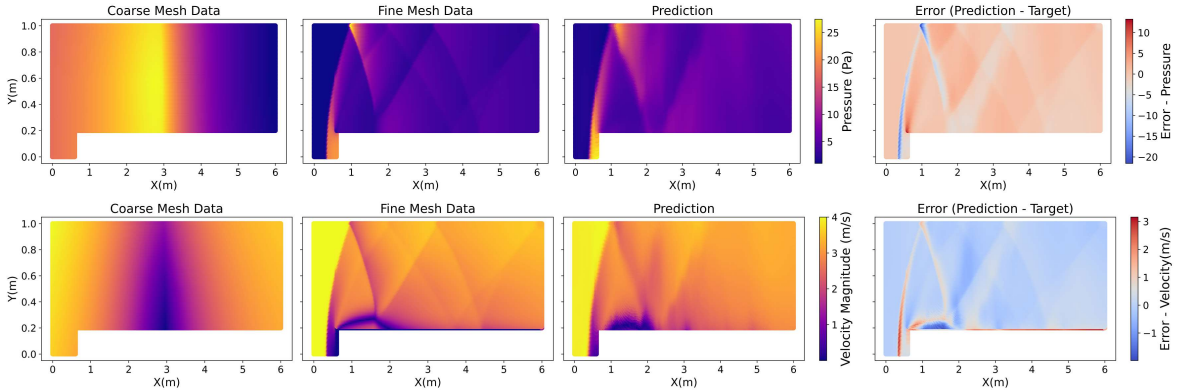



Figure 10: Scenario 2: Training on AR 3 and 4,and pressure and velocity prediction at t= 3.5s for inlet velocity 4 m/s for an AR 6 dataset.

## A.2 Case Study 2: Lid-driven Cavity

In this case study, we explore the lid-driven cavity, a relatively straightforward scenario compared to others, well known as a benchmark problem in computational fluid dynamics (CFD). The problem entails modeling the fluid flow within a cubic cavity, with a width of 1m and a lid velocity ($U_{lid}$) of 1m/s, resulting in intricate fluid phenomena, notably the formation of counter-rotating vortices at the cavity's bottom. The flow characteristics vary depending on factors such as the Reynolds number and aspect ratio. To simulate this, we

employed a transient solver known as *pisoFoam*, implementing the PISO algorithm. For the present work, it is focused on the quasi-steady state flow within the cavity.

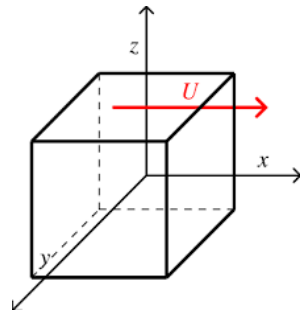
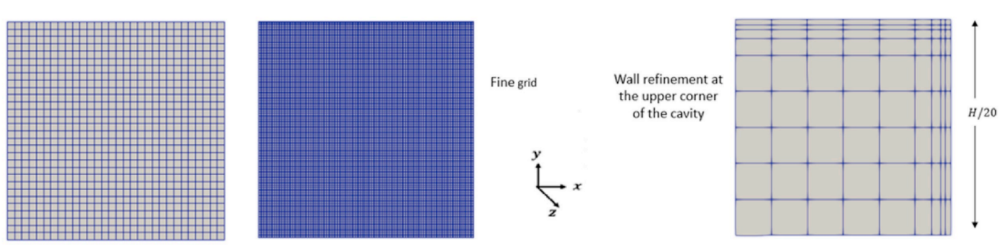



Figure 11: Lid Driven Cavity

### A.2.1 Problem Description

As explained above, the computational domain comprises a 3D cube cavity with a width of 1m, as illustrated in Figure 11. The cavity's height adjusts proportionally based on the aspect ratio. In this study, varying Reynolds numbers are achieved by altering the kinematic viscosity while maintaining a constant $U_{lid}$ value. Specifically, Reynolds numbers of 6000, 8000, 10000, and 12000 are considered, with grid sizes of 1/20, 1/30, and 1/40 employed for simulating coarse mesh data. For fine mesh data, a grid size of 1/120 is utilized. To enhance turbulence capture within the cavity, wall refinement is implemented in the fine mesh data's wall region, as depicted in the Figure 12, whereas such refinement is omitted for coarse data.

Following the generation of fine mesh data, it is overlaid onto the coarse mesh, resulting in identical mesh sizes denoted as $(r, d)$, where $r$ signifies the number of points, and $d$ represents the number of features. In this specific instance, the model's up-sampling aspect is bypassed since the input and output sizes match. In alignment with the source paper's methodology, which evaluated the model's predictability across diverse scenarios, including Reynolds number extrapolation, our study adheres to a similar approach. The model is trained on a subset of Reynolds numbers (e.g., 6000, 80000, 10000) and tested on entirely different ones (e.g., 12000), replicating six such scenarios. Herein, we concentrate on the initial four scenarios to showcase our model's versatility.




(a) Coarse Mesh (b) Fine Mesh along with wall refinement

Figure 12: The above figures depict the computational grids utilized in this work

**Results:** To demonstrate the model's effectiveness and predictive ability, we adhere to the protocol outlined in the source paper (Hanna et al. (2017)). In the six scenarios, we will execute the first four scenarios. The **"Scenario-1" (Reynolds number interpolation)** was showcased in the main paper. In this section, we will delve into the other scenarios. In these scenarios, the model's hyper-parameters consist of **r**, representing the radius of the graph, and **lr**, indicating the learning rate. Following several iterations, it was observed that the most effective values for "k" and "lr" are 0.005 and 0.001, respectively. Furthermore, each model undergoes training for 300 epochs.

**Scenario - 2:** This scenario illustrates Reynolds number extrapolation, where training and validation are conducted for flow at **Re - 6000, 8000, 10000**, and the model is subsequently tested at a different Reynolds number, **12000**. The grid size (1/30) and aspect ratio (1) remain constant. Figure 13 showcases the model's predictive prowess, with velocity contours plotted. The model accurately captures the turbulent nature of the flow, achieving an MSE of 3.5e-4 within just 120 seconds. Figure 14 showcase the training and validation loss.

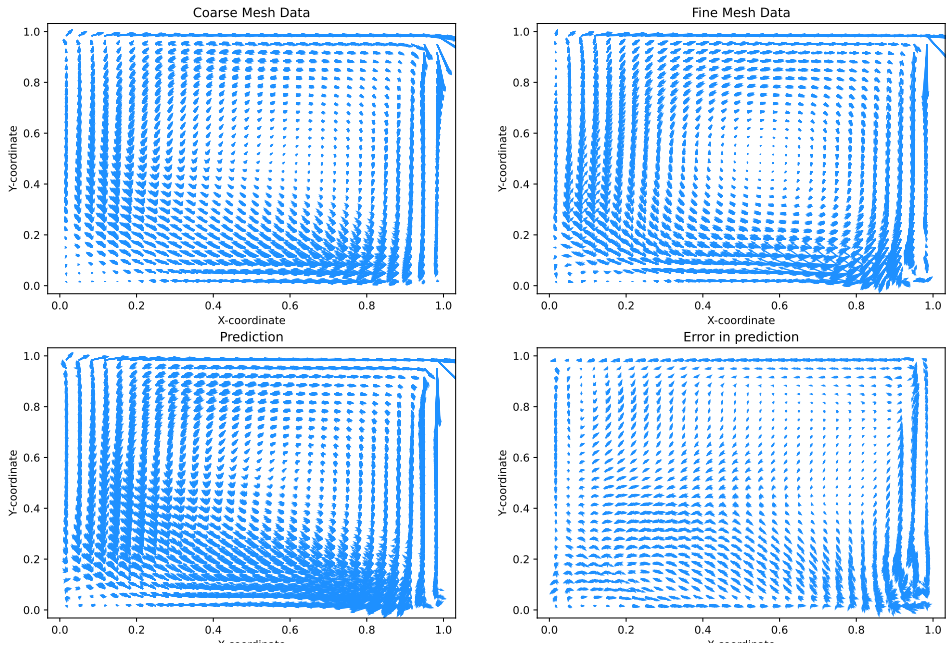



Figure 13: Velocity Contour Plot($U_x$ and $U_y$): Scenario - 2

In this case, since the grid size and the aspect ratio remain constant, the number of points in the input/output point cloud remains constant, i.e., 27,000. However, if we alter either of them, the number of points changes. In the next two scenarios, the grid size is modified, leading to different point cloud dimensions for the training and testing datasets.

In **Scenario - 3**, Reynolds number and Grid Size interpolation are employed, where training and validation encompass flow conditions at **Re - 8000, 12000 and Grid size - 1/40, 1/20**. The model is then assessed with a different parameter set, **Re - 10000 and Grid Size - 1/30**. The point cloud dimensions for training, validation, and testing datasets are 64,000, 8,000, and 27,000, respectively, showcasing the model's adaptability to

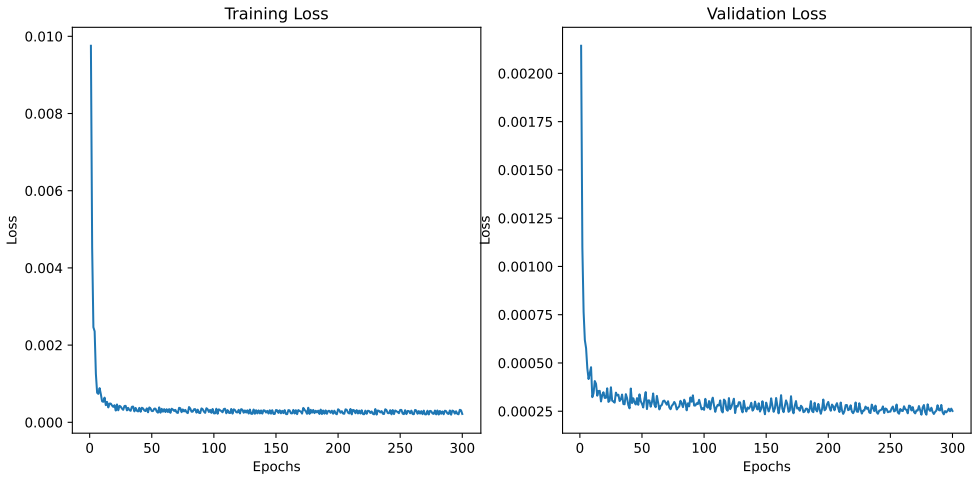



Figure 14: Training and Validation Loss

various mesh dimensions. Figure 15 illustrates the velocity contour plot, highlighting the model's adeptness in accurately capturing turbulence, achieving an MSE of 3e-4 within a mere 156 seconds. Figure 16 showcase the training and validation loss.

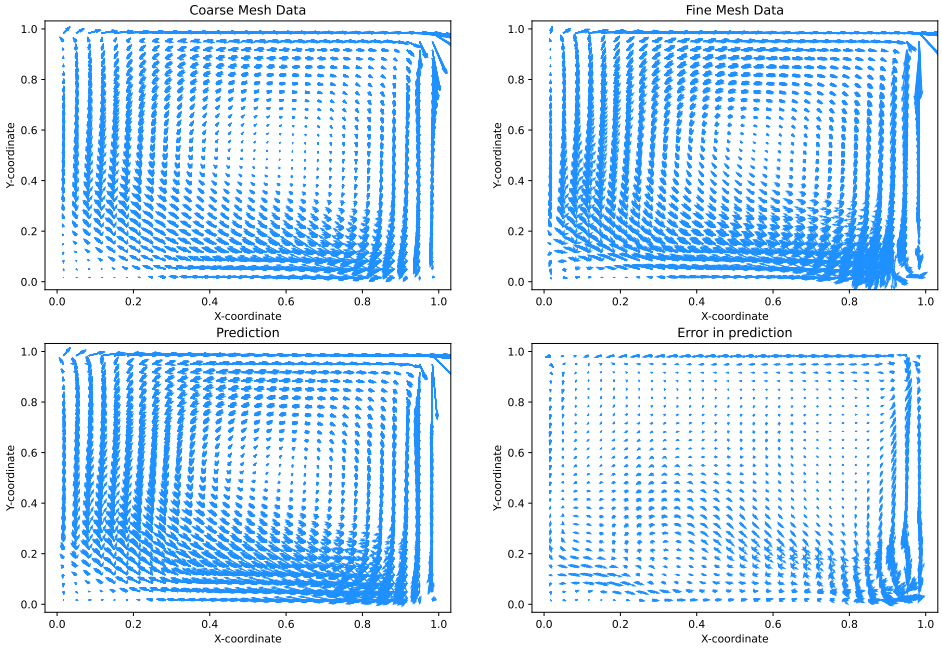



Figure 15: Velocity Contour Plot($U_x$ and $U_y$): Scenario - 3

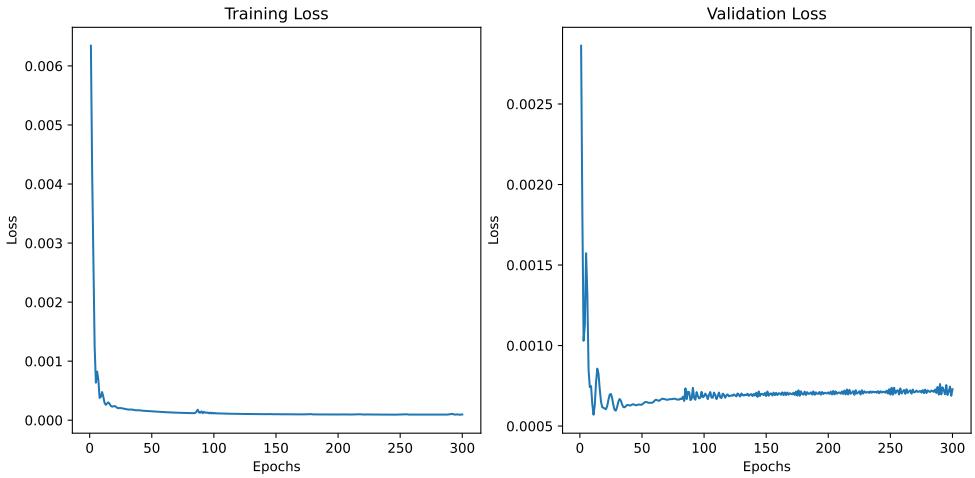



Figure 16: Training and Validation Loss

**Scenario - 4:** In this scenario, both Reynolds number and grid size are varied for training and validation, covering flow conditions at **Re - 8000, 10000 and Grid size - 1/30, 1/20**. The model is then tested with a different parameter set, **Re - 12000 and Grid Size - 1/40**. The point cloud dimensions for training, validation, and testing datasets are 64,000, 8,000, and 27,000, respectively. This scenario presents a greater challenge compared to the previous one. Figure 18 demonstrates the model's efficacy in accurately capturing turbulence with an MSE of 2e-4 within just 50 seconds. Figure 17 showcase the training and validation loss.

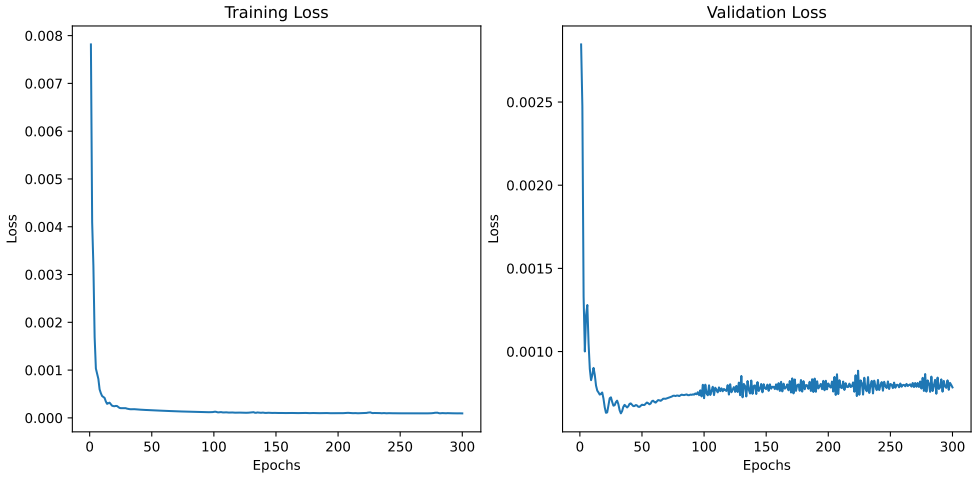



Figure 17: Training and Validation Loss

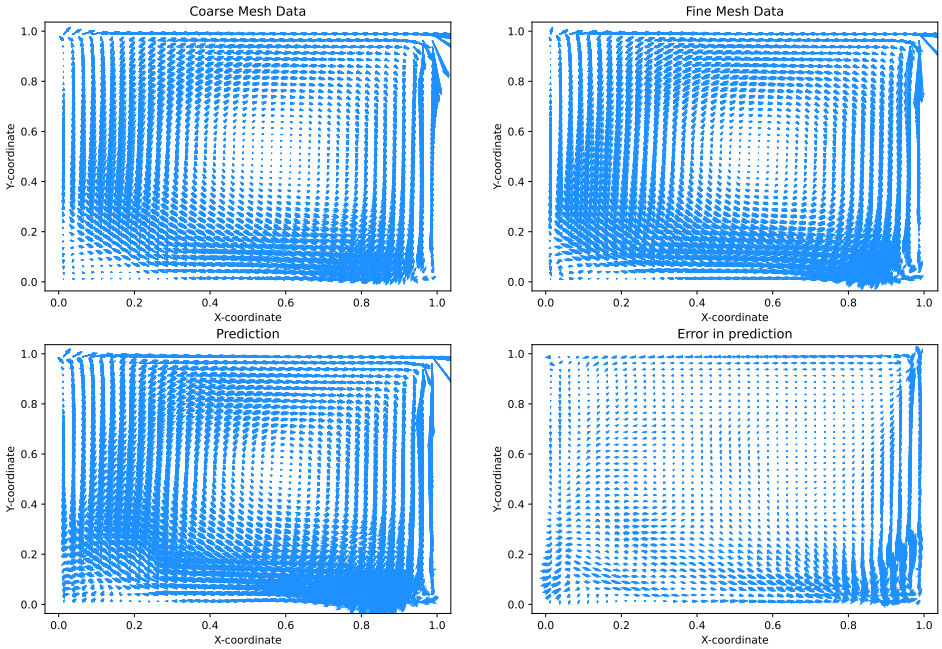



Figure 18: Velocity Contour Plot($U_x$ and $U_y$): Scenario - 4