Redefining Super-Resolution: Fine-mesh PDE predictions without classical simulations

Rajat Kumar Sarkar[*]
Researcher
TCS Research
rajat.sarkar1@tcs.com

Ritam Majumdar
Researcher
TCS Research
ritam.majumdar@tcs.com

Vishal Jadhav
Scientist
TCS Research
vi.suja@tcs.com

Sagar Srinivas Sakhinana
Scientist
TCS Research
sagar.sakhinana@tcs.com

Venkataramana Runkana
Chief Scientist
TCS Research
venkat.runkana@tcs.com

Abstract

In Computational Fluid Dynamics (CFD), coarse mesh simulations offer computational efficiency but often lack precision. Applying conventional super-resolution to these simulations poses a significant challenge due to the fundamental contrast between downsampling high-resolution images and authentically emulating low-resolution physics. The former method conserves more of the underlying physics, surpassing the usual constraints of real-world scenarios. We propose a novel definition of super-resolution tailored for PDE-based problems. Instead of simply downsampling from a high-resolution dataset, we use coarse-grid simulated data as our input and predict fine-grid simulated outcomes. Employing a physics-infused UNet upscaling method, we demonstrate its efficacy across various 2D-CFD problems such as discontinuity detection in Burger's equation, Methane combustion, and fouling in Industrial heat exchangers. Our method enables the generation of fine-mesh solutions bypassing traditional simulation, ensuring considerable computational saving and fidelity to the original ground truth outcomes. Through diverse boundary conditions during training, we further establish the robustness of our method, paving the way for its broad applications in engineering and scientific CFD solvers.

1 Introduction

Computational Fluid Dynamics (CFD) plays a crucial role in comprehending intricate physical phenomena spanning scientific and engineering domains, including aerospace [22][25], automotive [27], energy[19], and more. To gain a profound understanding of these physical phenomena, it becomes imperative to conduct simulations at high mesh resolutions of the governing equation of fluid flow like the Navier-Stokes equation, encompassing a broad spectrum of fluid structures, ranging from large-scale patterns[18][14] to subgrid-scale features like small eddies within fluid flow systems [16][24]. However, simulation of fluid flow within intricate geometries at high mesh resolutions is inherently computationally intensive and time-consuming[23]. This has led to the popularity of coarse mesh simulations, primarily due to their computational efficiency. Nevertheless, coarse mesh simulations deal with a persistent challenge - their inherent limitation of low mesh resolution, often compromising the precision of the results obtained. Addressing this resolution disparity becomes paramount in accurately capturing the subtleties of fluid dynamics.

[*] Corresponding author

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

In order to circumvent the issue of simulating high-resolution mesh solutions of PDEs, several researchers have adapted the super-resolution technique, commonly used in Computer Vision[10][2][28] to map low-resolution data to high-resolution data using a supervised loss. This approach involves an initial downsampling of the original high-resolution data to low resolution, accomplished through various downsampling techniques such as max-pooling, average-pooling, and nearest-neighbor methods. Subsequently, they leverage a combination of numerous variants of deep learning architectures, including multi-layer perceptron [5, 20], convolutional neural networks [7, 21, 13, 32, 6], and generative adversarial networks [29, 3, 11, 9, 31, 4]. These techniques are employed to reconstruct the original high-resolution data from the downsampled low-resolution data they have generated. Moreover, some researchers have integrated physics-based principles into their networks to effectively capture the underlying physics within the low-resolution data, enhancing the reconstruction of high-resolution data [8, 3, 4, 6, 1].

However, high-resolution mesh solutions of partial differential equations (PDEs) contain an extensive amount of information regarding the underlying physics within the mesh stencil. In contrast, low-resolution mesh stencils tend to exhibit irregularities in capturing these governing physics. Applying the conventional definition of super-resolution directly to the world of PDEs presents two notable challenges. Firstly, the process of generating a dataset by downsampling high-resolution data to create low-resolution data differs significantly from generating simulations of low-resolution data using conventional numerical solvers. Secondly, there is a shortage of high-resolution mesh data, making it difficult to acquire sufficient training data for super-resolution models. This distinction arises from the complex physical properties that are inherently associated with high-resolution data. Simply downsampling from high-resolution data results in retaining the majority of the governing physics(refer to 5.5), which may not accurately reflect real-world scenarios. To address this challenge, we have devised an alternative approach for framing the super-resolution problem to align more closely with practical applications. Instead of downsampling high-resolution data, we utilize coarse-grid simulated data as input and predict fine-grid simulated data as the super-resolution output. This approach emulates real-world conditions accurately and overcomes the limitations of conventional super-resolution techniques.

In summary, our physics-informed UNet upscaling approach effectively predicts fine mesh data features from coarse mesh data, demonstrating its effectiveness across various CFD datasets. Enabling the generation of high-fidelity fine-mesh solutions without the need for traditional time-consuming CFD simulations, our method offers both significant time savings and accuracy. We've demonstrated its robustness through diverse boundary conditions during training, highlighting its potential for widespread applications in engineering and scientific CFD solvers.

2 Methodology

2.1 Problem formulation

Consider a training dataset consisting of pairs of coarse mesh data and fine mesh data represented as (C, F). The primary goal of this research is to train an inductive supervised model ($f : C \rightarrow F$) to effectively capture and map the non-linear relationship between coarse mesh data $C(x_1, x_2, ...x_n) \in R^m$ and fine mesh data $F(x_1, x_2, ...x_n) \in R^d$ of n features, where $m \ll d$. Mapping learns the characteristics of the system variables and properties described by governing equations. The model takes coarse mesh data at a given time step t and predicts the corresponding fine mesh data, effectively performing upsampling across features (x_n) as shown in table 2.

2.2 Model Architecture

The architecture of our Physics Informed UNet (PIUNet) for upscaling coarse-grid fluid flow simulation data is depicted in Figure 1. PIUNet bridges the gap between low-resolution coarse mesh data and high-resolution fine mesh data, enabling accurate fluid flow behavior predictions. The PIUNet begins with four contracting convolution layers. Each layer comprises a 3x3 double unpadded convolution operation, followed by ReLU activation, and subsequently a 2x2 max-pooling layer. These layers progressively downsample the input coarse grid data, allowing the network to capture increasingly abstract and high-level features. It is followed by a bottleneck layer capturing high-level features while preserving spatial information. Following the bottleneck layer, four expansion convolution layers are employed. Each layer consists of a deconvolution operation, a skip connection, and another

3x3 double unpadded convolution operation, followed by ReLU activation. The skip connection guides the upsampling process, retaining essential spatial information. After the expansion layers, a 1x1 convolution layer reconstructs the output to match the dimension of the coarse mesh data. A 1x1 convolution layer precedes the bilinear upsampler, which leverages the learned representations from the network to upsample the data to the fine-mesh dimension effectively. The detailed UNet architecture is visually represented in Figure 2. In the training process, we employ a customized loss function that combines both data loss and physics loss. This loss function encourages and updates the gradient of the network to learn not only from the data but also from the underlying physics of the system.

2.3 Loss Function:

In the training process, we utilize a customized loss function that combines data and physics components. The data loss ($\mathcal{L}_{data}$) is computed as the Mean Square Error (MSE) between fine mesh and predicted fine mesh data, while physics loss ($\mathcal{L}_{physics}$) measures the MSE of the convective and diffusive terms from the governing equations. The total loss function is given by:

Total Loss Function: $$\mathcal{L}_{total} = \mathcal{L}_{data} + \mathcal{L}_{physics} \tag{1}$$

The physics loss is tailored for different datasets with distinct weights, the details of these weights and the equation of the physics loss for each respective dataset can be found in the Appendix 5.4.

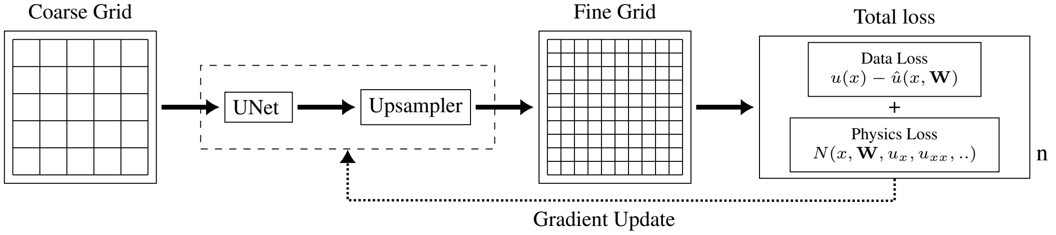


Figure 1: PIUNet Architecture

3 Results and Discussion

In this section, we present the outcomes of our experiments, focusing on the performance evaluation of the PIUNet across various datasets as previously mentioned. Detailed results are summarized in Table 1, providing comprehensive insights into our findings. We employed two widely recognized interpolation techniques, namely Bi-linear Interpolation and Bi-cubic Interpolation as baseline methods to compare the super-resolution capabilities for upscaling coarse mesh data to fine mesh data features. Initially, we utilized a vanilla UNet-based super-resolution approach. As we sought to improve the results, we integrated the physics loss, resulting in the development of PIUNet. We have applied this to three CFD problems, phenomena represented by Burger's Equation, fouling in industrial heat exchangers, and methane combustion.

Burger's Equation:[17] Comparing the UNet-based super-resolution approach with baseline interpolations, we observe significant improvements. In terms of RMSE, we achieve approximately 18x and 14x enhancements for U_x and U_y in the x and y directions, respectively. Additionally, MAE demonstrates improvements of roughly 30x for U_x and 32x for U_y. The introduction of physics into UNet, i.e., the Physics-Informed UNet, further enhances results by reducing RMSE by approximately 27% for U_x and 27% for U_y.

Industrial Heat Exchanger:[15] This problem involves six features, namely fluid temperatures T_1,T_2 and T_3, and matrix temperatures T_{m_1},T_{m_2},T_{m_3}. Similar to Burger's equation, the UNet-based super-resolution method outperforms baseline interpolations. The Physics-Informed UNet further enhances results, achieving a 36.8% reduction in RMSE for T_1, 30.7% for T_2 and T_3, 35.04% for T_{m_1}, and 20.06% for T_{m_2},T_{m_3}. MAE also witnesses improvements of approximately 37% for T_1, 50% for T_2,T_3, 39.2% for T_{m_1}, and 27.48% for T_{m_2} and T_{m_3}.

Methane Combustion:[30] This problem involves seven outputs, including adiabatic flame temperature (T_{adia}), x-direction velocity, y-direction velocity, and mass fractions of species CH_4, O_2,

H_2O, and CO_2. Notably, the rise in adiabatic flame temperature and the depletion of the CH_4 mass fraction significantly influence combustion. The Physics-Informed UNet-based super-resolution method outperforms baseline interpolation methods, with RMSE improvements of 73% in T_{adia} and 20.47% in the mass fraction of CH_4. For other features such as velocities and mass fractions of O_2, H_2O, and CO_2, results align closely with baseline interpolation due to minimal differences between coarse and fine mesh data.

Table 1: Performance Comparison of PIUNet vs. Baseline Interpolations on test data using RMSE, MAE, and R^2 Metrics for Diverse Datasets(Best scores are in bold)

	Features	Algorithm	RMSE	MAE	R^2		Features	Algorithm	RMSE	MAE	R^2
Burger's Eq	51×51 → 401×401					**Methane Combustion**	50×20 → 500×100				
	X-Velocity (U_x)	Bilinear	0.4927	0.3858	-0.3368		Temperature (T_{adia})	Bilinear	78.317	32.444	0.9776
		Bicubic	0.5133	0.3927	-0.4513			Bicubic	77.374	31.486	0.9779
		UNet	0.0283	0.0126	**0.9955**			UNet	30.718	13.224	0.9963
		PIUNet	**0.0207**	**0.0062**	0.9948			**PIUNet**	**20.954**	**10.385**	**0.9984**
	Y-Velocity (U_y)	Bilinear	0.3517	0.2602	-0.4616		X-Velocity (U_x)	Bilinear	0.0289	0.0122	0.9843
		Bicubic	0.3804	0.2752	-0.7106			Bicubic	0.0290	0.0121	0.9842
		UNet	0.0247	0.00817	0.9927			UNet	0.0296	0.0177	0.9839
		PIUNet	**0.0225**	**0.00289**	**0.9965**			**PIUNet**	**0.0286**	**0.0164**	**0.9862**
Industrial Heat Exchanger	30×30 → 480×480						Y-Velocity (U_y)	Bilinear	0.0325	**0.0118**	0.9819
	Gas flow Temperature (T_1)	Bilinear	3.1859	2.4508	0.9971			Bicubic	0.0332	0.0125	0.9815
		Bicubic	3.2192	2.4635	0.9971			UNet	0.0324	0.0152	0.9830
		UNet	2.5920	2.0306	0.9940			**PIUNet**	**0.0324**	0.0158	**0.9835**
		PIUNet	**2.0112**	**1.5386**	**0.9985**		Mass Fractions				
	Air flow Temperature (T_2 and T_3)	Bilinear	4.0527	3.4445	0.9980		CH_4	Bilinear	0.0356	0.0104	0.9920
		Bicubic	4.0008	3.4324	0.9980			Bicubic	0.0357	0.0105	0.9920
		UNet	3.7728	2.8390	0.9981			UNet	0.0321	0.0140	0.9938
		PIUNet	**2.8084**	**1.7214**	**0.9988**			**PIUNet**	**0.0283**	**0.0138**	**0.9954**
	Gas Matrix Temperature (T_{m_1})	Bilinear	3.0919	2.6329	0.9981		O_2	Bilinear	0.0105	0.0028	0.9889
		Bicubic	3.1228	2.6244	0.9981			Bicubic	**0.0104**	**0.0027**	**0.9891**
		UNet	2.5787	2.0840	0.9979			UNet	0.0116	0.0059	0.9870
		PIUNet	**2.0084**	**1.5985**	**0.9987**			**PIUNet**	0.0106	0.0030	0.9888
	Air Matrix Temperature (T_{m_2})	Bilinear	3.6404	3.1354	0.9979		H_2O	Bilinear	0.0038	0.0014	0.9893
		Bicubic	3.5790	3.1288	0.9980			Bicubic	**0.0037**	**0.0013**	**0.9899**
		UNet	4.2434	3.6032	0.9971			UNet	0.0076	0.0037	0.9598
		PIUNet	**2.9098**	**2.2735**	**0.9982**			**PIUNet**	0.0040	0.0020	0.9891
	Air Matrix Temperature (T_{m_3})	Bilinear	3.6404	3.1354	0.9979		CO_2	Bilinear	0.0052	0.0019	0.9844
		Bicubic	3.5790	3.1288	0.9980			Bicubic	0.0052	0.0018	**0.9849**
		UNet	4.2434	3.6032	0.9971			UNet	0.0094	0.0058	0.9564
		PIUNet	**2.9098**	**2.2735**	**0.9982**			**PIUNet**	**0.0051**	**0.0018**	0.9844

Furthermore, as evident in Table 2, our utilization of the Physics-Informed UNet in conjunction with the physics model has resulted in a substantial reduction in computational costs for fine mesh simulations.

Table 2: Accelerated CFD Simulations: PIUNet's speedup compared to traditional techniques

CFD problem	Simulation technique	Grid size	Simulation Time (s)
2D Burger's Equation	Finite Difference Method	51 × 51	151
	(MATLAB)	401 × 401	3623
	FDM + PIUNet trained model	401 × 401	**152.764** (**≈24X speed**)
Counterflow Methane Combustion	Finite Volume Method	50 × 20	74.59
	(Open FOAM)	500 × 100	7146.67
	FDM + PIUNet	500 × 100	**76.795** (**≈93X speed**)
Industrial Heat Exchanger	Finite Difference Method	30 × 30	6.969
	(Python)	480 × 480	2200.45
	FDM + PIUNet	480 × 480	**9.369** (**≈235X speed**)

4 Conclusion

In conclusion, our study highlights the effectiveness of the PIUNet in improving super-resolution tasks of upsampling the coarse grid data to fine grid data across different datasets. For Burger's equation, we achieved substantial RMSE and MAE enhancements, especially for fluid velocity in the x and y directions, by incorporating physics into the UNet. In the industrial heat exchanger scenario, PIUNet outperformed baseline methods significantly reducing RMSE and MAE for various temperature parameters. In the case of methane combustion, PIUNet notably improved RMSE for adiabatic flame temperature and CH_4 mass fraction. These results demonstrate PIUNet's potential in capturing complex physical behaviors in different boundary conditions, with promising applications in the fluid dynamics of industrial systems.

Impact Statement

The current state-of-the-art super-resolution techniques involve reconstructing high-resolution data from downsampled low-resolution data, and face challenges when applied to traditional CFD simulations. These challenges stem from the scarcity of high-resolution mesh data for training, and from the fundamental differences between downsampling high-resolution data and simulating low-resolution data using traditional numerical solvers. This distinction arises from the complex physical properties that are inherently associated with high-resolution data. Simply downsampling from high-resolution data results in retaining the majority of the governing physics, which may not accurately reflect real-world scenarios. We utilize coarse-grid data as the input to predict fine-grid results in super-resolution, seamlessly integrating it into traditional CFD models. This eliminates the need for resource-intensive fine-mesh CFD simulations, significantly reducing computation time while preserving intricate fluid behavior details. This advancement can enhance the accuracy of computational simulations, leading to improved designs and operations optimization in industries such as aerospace, energy, etc. However, it's important to acknowledge the limitation of our current work, which lies in the specific grid upsampling training. As a future endeavor, we plan to modify our model to overcome this limitation and extend its applicability to irregular and complex geometry meshes.

Acknowledgment

The author extends his appreciation to TCS Research for the opportunity to undertake this project. A special acknowledgment goes to co-authors Ritam Majumdar, Vishal Jadhav, Dr.Sagar Sakhinanana, and Dr.Venkataramana Runkana for their consistent support, guidance, and invaluable insights throughout the research process. Gratitude is also extended to colleagues Shivam Gupta and Krishna Sai Sudhir Aripirala for their assistance in generating the dataset, a pivotal aspect of the project. Additionally, thanks are conveyed to the organizers and reviewers of the NeurIPS ML4PS workshop for providing a platform to disseminate our research findings. Finally, the author expresses thanks to family and friends for their encouragement and understanding throughout this academic pursuit.

References

[1] Rajat Arora. Physrnet: Physics informed super-resolution network for application in computational solid mechanics. In *2022 IEEE/ACM International Workshop on Artificial Intelligence and Machine Learning for Scientific Applications (AI4S)*, pages 13–18. IEEE, 2022.

[2] Vivek Bannore and Vivek Bannore. *Iterative-interpolation super-resolution (IISR)*. Springer, 2009.

[3] Mathis Bode, Michael Gauding, Konstantin Kleinheinz, and Heinz 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.

[4] Mathis Bode, Michael Gauding, Zeyu Lian, Dominik Denker, Marco Davidovic, Konstantin Kleinheinz, Jenia Jitsev, and Heinz 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.

[5] N Benjamin Erichson, Lionel Mathelin, Zhewei Yao, Steven L Brunton, Michael W Mahoney, and J Nathan Kutz. Shallow neural networks for fluid flow reconstruction with limited sensors. *Proceedings of the Royal Society A*, 476(2238):20200097, 2020.

[6] Soheil Esmaeilzadeh, Kamyar Azizzadenesheli, Karthik Kashinath, Mustafa Mustafa, Hamdi A Tchelepi, Philip Marcus, Mr Prabhat, Anima 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.

[7] Kai Fukami, Koji Fukagata, and Kunihiko Taira. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019.

[8] Han Gao, Luning Sun, and Jian-Xun Wang. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, 33(7), 2021.

[9] Alejandro Güemes, Stefano Discetti, Andrea Ianiro, Beril Sirmacek, Hossein Azizpour, and Ricardo Vinuesa. From coarse wall measurements to turbulent velocity fields through deep learning. *Physics of Fluids*, 33(7), 2021.

[10] Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991.

[11] Hyojin Kim, Junhyuk Kim, Sungjin Won, and Changhoon Lee. Unsupervised deep learning for super-resolution reconstruction of turbulence. *Journal of Fluid Mechanics*, 910:A29, 2021.

[12] Chung-Hsiung Li. A numerical finite difference method for performance evaluation of a periodic-flow heat exchanger. *Journal of Heat Transfer*, 1983.

[13] Bo Liu, Jiupeng Tang, Haibo Huang, and Xi-Yun Lu. Deep learning methods for super-resolution reconstruction of turbulent flows. *Physics of Fluids*, 32(2), 2020.

[14] Yangwei Liu, Xianjun Yu, and Baojie Liu. Turbulence models assessment for large-scale tip vortices in an axial compressor rotor. *Journal of Propulsion and Power*, 24(1):15–25, 2008.

[15] Ritam Majumdar, Vishal Jadhav, Anirudh Deodhar, Shirish Karande, Lovekesh Vig, and Venkataramana Runkana. Real-time health monitoring of heat exchangers using hypernetworks and pinns. *arXiv preprint arXiv:2212.10032*, 2022.

[16] PJ Mason and NS Callen. On the magnitude of the subgrid-scale eddy coefficient in large-eddy simulations of turbulent channel flow. *Journal of Fluid Mechanics*, 162:439–462, 1986.

[17] Marlon S Mathias, Wesley P de Almeida, Jefferson F Coelho, Lucas P de Freitas, Felipe M Moreno, Caio FD Netto, Fabio G Cozman, Anna Helena Reali Costa, Eduardo A Tannuri, Edson S Gomi, et al. Augmenting a physics-informed neural network for the 2d burgers equation by addition of solution data points. In *Brazilian Conference on Intelligent Systems*, pages 388–401. Springer, 2022.

[18] Elia Merzari, Aleks Obabko, Paul Fischer, Noah Halford, Justin Walker, Andrew Siegel, and Yiqi Yu. Large-scale large eddy simulation of nuclear reactor flows: Issues and perspectives. *Nuclear Engineering and Design*, 312:86–98, 2017.

[19] Aaron Miller, Byungik Chang, Roy Issa, and Gerald Chen. Review of computer-aided numerical simulation in wind energy. *Renewable and sustainable energy Reviews*, 25:122–134, 2013.

[20] Nirmal J Nair and Andres Goza. Leveraging reduced-order models for state estimation using deep learning. *Journal of Fluid Mechanics*, 897:R1, 2020.

[21] Octavi Obiols-Sales, Abhinav Vishnu, Nicholas P Malaya, and Aparna Chandramowlishwaran. Surfnet: Super-resolution of turbulent flows with transfer learning using small datasets. In *2021 30th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pages 331–344. IEEE, 2021.

[22] Pradeep Raj. Requirements for effective use of cfd in aerospace design. In *NASA CONFERENCE PUBLICATION*, pages 15–15. NASA, 1995.

[23] F Roelofs and A Shams. Cfd—introduction. In *Thermal Hydraulics Aspects of Liquid Metal Cooled Nuclear Reactors*, pages 213–218. Elsevier, 2019.

[24] Hamid Sarlak, C Meneveau, and Jens Nørkær Sørensen. Role of subgrid-scale modeling in large eddy simulation of wind turbine wake interactions. *Renewable Energy*, 77:386–399, 2015.

[25] KP Singh, JS Mathur, V Ashok, and Debasis Chakraborty. Computational fluid dynamics in aerospace industry in india. *Defence Science Journal*, 60(6), 2010.

[26] T Skiepko. The effect of matrix longitudinal heat conduction on the temperature fields in the rotary heat exchanger. *International journal of heat and mass transfer*, 31(11):2227–2238, 1988.

[27] Kamil Szewc, Jona Mangold, Christoph Bauinger, Martin Schifko, and Chong Peng. Gpu-accelerated meshless cfd methods for solving engineering problems in the automotive industry. Technical report, SAE Technical Paper, 2018.

[28] Jing Tian and Kai-Kuang Ma. A survey on super-resolution imaging. *Signal, Image and Video Processing*, 5:329–342, 2011.

[29] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018.

[30] Qi Yang, Peng Zhao, and Haiwen Ge. reactingfoam-sci: An open source cfd platform for reacting flow simulation. *Computers & Fluids*, 190:114–127, 2019.

[31] Mustafa Z Yousif, Linqi Yu, and Hee-Chang Lim. High-fidelity reconstruction of turbulent flow from spatially limited data using enhanced super-resolution generative adversarial network. *Physics of Fluids*, 33(12), 2021.

[32] Xu-Hui Zhou, James E McClure, Cheng Chen, and Heng Xiao. Neural network–based pore flow field prediction in porous media using super resolution. *Physical Review Fluids*, 7(7):074302, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?[YES]
   (b) Did you describe the limitations of your work?[YES] We comment on the limitations in the impact statement section.
   (c) Did you discuss any potential negative societal impacts of your work? [NO]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them?[YES]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results?[N/A]
   (b) Did you include complete proofs of all theoretical results?[NO]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)?[YES]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?[YES]. Each experiment was run five times.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)?[YES]. Please refer to the Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [YES].
   (b) Did you mention the license of the assets?[N/A].
   (c) Did you include any new assets either in the supplemental material or as a URL? [NO].
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating?[N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [NO].
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable?[N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable?[N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation?[N/A]

5 Appendix

5.1 Training and Hyperparameter details

In our experiments, we conducted training using the PyTorch framework on a Nvidia®P100 GPU with 16GB of memory. We generated three distinct two-dimensional datasets for Burger equation simulation, Counterflow methane combustion simulation, and industrial heat exchanger simulation.. For the 2D-Burger dataset, we produced 10 boundary conditions, each comprising 200-time steps of coarse mesh data (51×51 grid) paired with fine mesh data (401×401 grid). In the 2D-counterflow methane combustion dataset, there were 10 boundary conditions, each spanning 50-time steps, with coarse (50×20) and fine (500×100) mesh data. The industrial heat exchanger dataset included 100 boundary conditions of steady state, with coarse mesh data (30×30 grid) and fine mesh data (480×480 grid). During training, we employed the Adam optimizer for 500 epochs, starting with a learning rate of $1e^{-3}$ and weight decay of $5e^{-4}$. The learning rate was reduced by 30% after 5 epochs without validation improvement. We determined the total dataset size for each case by multiplying the number of boundary conditions by the number of time steps. Data was partitioned into training, validation, and testing sets using a 60/20/20 ratio for evaluation. All the distinct weights of the physics loss of the different datasets are the hyperparameters of our model.

5.2 UNet Architecture

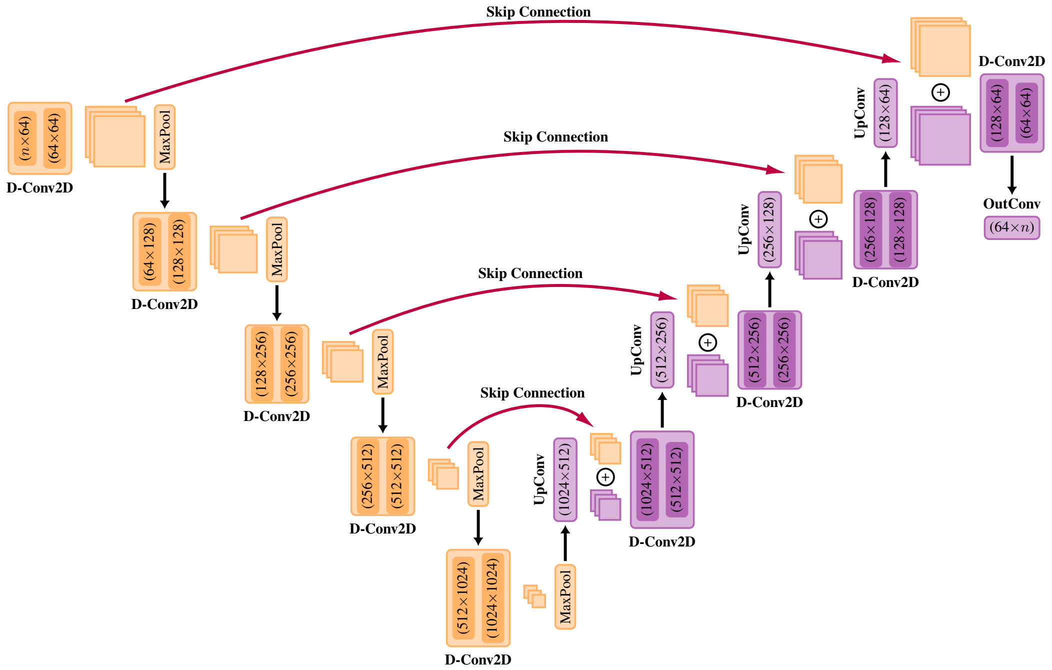


Figure 2: UNet Architecture

5.3 Dataset Information

Burger's Equation We investigate the behavior of wave phenomena in a viscous fluid flow by considering the 2D Burger's equation. This equation is a fundamental non-linear advection-diffusion problem that characterizes the dynamics of wave-like structures in a two-dimensional fluid medium. The 2D Burger's Equation is mathematically expressed as follows:

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \left(\nabla^2 \mathbf{u} \right) \tag{2}$$

Here, the variable $\mathbf{u}$ represents the velocity field and ν represents kinematic viscosity. For our investigations, we have adopted sinusoidal initial conditions, $U_x(0, x, y) = \sin(2\pi x)\sin(2\pi y)$ and $U_y(0, x, y) = \sin(\pi x)\sin(\pi y)$. These initial conditions, as detailed in [17], serve as the starting point for our experiments. Specifically, we explore a range of Dirichlet boundary conditions for $\mathbf{u}$ within the intervals (0, 1)m/s to gain insights into the diverse behaviors exhibited by the system. The 2D-Burger simulation dataset was generated in a rectangular domain using MATLAB-based code[17]. This code approximated spatial and time derivatives using 6th-order FDM and 4th-order Runge-Kutta schemes with a time step (Δt) of 10^{-5}

Methane Combustion We investigate the fundamental chemical reaction of methane combustion within a 2D-Laminar counter-flow configuration. This reaction is pivotal in various combustion systems, such as gas turbines and furnaces. In the counter-flow setup, the velocities of methane (the fuel) and air are oriented in opposing directions. The chemical reaction governing the combustion of methane, $\mathrm{CH_4}$, with air is as follows:

$$\mathrm{CH_4} + 2\mathrm{O_2} \rightarrow \mathrm{CO_2} + 2\mathrm{H_2O}$$

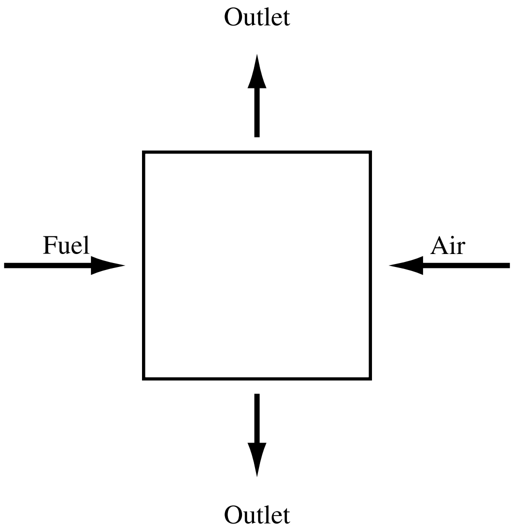


Figure 3: Computational Domain of a 2D-counterflow methane combustion

Our analysis is based on a set of governing equations that describe the key physical aspects of this combustion process:

$$\begin{aligned}
&\text{continuity eq:} && \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0 \\
&\text{momentum eq:} && \partial_t(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \mu(\nabla^2 \mathbf{u}) \\
&\text{species transport eq:} && \partial_t \left(\rho Y_i\right) + \nabla \cdot (\rho \mathbf{u} Y_i) = \mu \nabla^2 Y_i + \dot{R}_i \\
&\text{energy eq:} && \partial_t(\rho h) + \nabla(\rho \mathbf{u} h) + \partial_t(\rho K) + \nabla(\rho \mathbf{u} K) - \partial_t p = \alpha \nabla^2 h + \dot{R}_{\text{heat}}
\end{aligned} \tag{3}$$

In these equations, $\mathbf{u}$ represents the velocity vector, ρ denotes density, p stands for pressure, and μ represents the dynamic viscosity of the fluid mixture. The species transport equation involves Y_i (mass fraction of species i) and $\dot{R}_i$ (production rate of species i). The energy equation incorporates h (internal energy), where α, K, and $\dot{R}_{\text{heat}}$ correspond to thermal diffusivity, kinematic energy, and heat generation due to the chemical reaction, respectively. Considering isobaric combustion, all pressure gradient terms become zero. Our investigation includes a range of Dirichlet boundary conditions for $\mathbf{u}$ for both fuel and air within the intervals (0.1, 0.6) m/s. This exploration provides valuable insights into the diverse behaviors exhibited by the combustion system. The simulation dataset was conducted within a 2cm×2cm domain by employing OpenFOAM reactingFoam solver[30] while assuming laminar flow conditions.

Industrial Heat exchanger We explore the non-dimensional, 2D formulation of counterflow heat transfer within an industrial heat exchanger. These heat exchangers play a vital role in thermal power plants, enhancing overall thermal efficiency. Monitoring the internal temperature profiles of these heat exchangers is crucial to prevent failures stemming from intricate thermal and chemical deposition phenomena. We use Eq 4 for heat conduction and Eq 5 for convective heat transfer. Our system has six outputs: three fluid temperatures (T) and three metal temperatures (T_m) at specific coordinates (φ, z). Given that this equipment operates under mostly consistent conditions, with minimal fluctuations in flow patterns, we have opted to utilize steady-state energy transfer equations to elucidate the heat transfer phenomena. It is crucial to emphasize that, in this context, the preservation of energy takes on great significance due to the notable variations in temperature gradients that arise from the processes of convection and conduction along the longitudinal axis of the matrix.[26].

$$\partial_\varphi T_{m_j} = NTU_{m_j}(T_j - T_{m_j}) + Pe_{m_j}^{-1} \partial_z^2 T_{m_j} \tag{4}$$

$$\partial_z T_j = NTU_{m_j}\left(T_{m_j} - T_j,\right) \quad j = 1, 2, 3 \tag{5}$$

The non-dimensional boundary conditions are set by gas($T_{in,1}$), primary air($T_{in,2}$), and secondary air inlet temperatures($T_{in,3}$) is given by,

$$\begin{aligned}
&T_j\left(\varphi, z=0\right) = T_{in,j}, \quad j=1,2,3 \\
&T_{m_1}(\varphi=0,z) = T_{m_3}(\varphi=1,1-z) \\
&T_{m_1}(\varphi=1,z) = T_{m_2}(\varphi=0,1-z) \\
&T_{m_2}(\varphi=1,z) = T_{m_3}(\varphi=0,z), \\
&\partial_z T_{m_j}[z=0,1] = 0, \quad j=1,2,3
\end{aligned} \tag{6}$$

the above matrix boundary conditions impose continuity constraints on metal temperatures. The simulation dataset is generated in a cylindrical domain as shown in figure 4 using finite difference method-based solver detailed in [12]

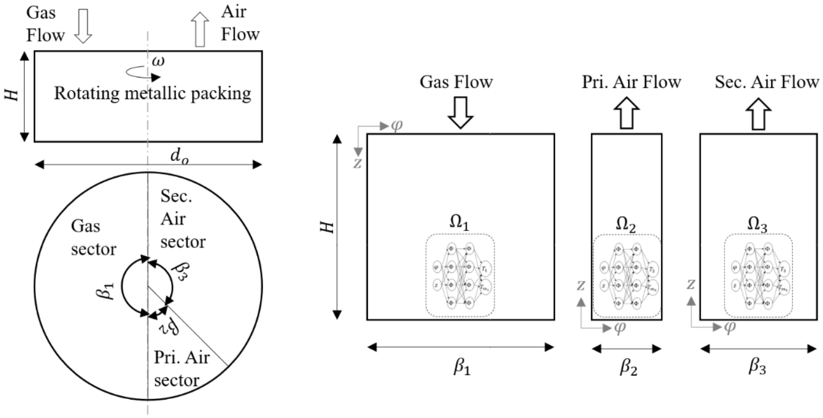


Figure 4: Computational Domain of an Industrial Heat Exchanger

5.4 Physics Loss Function:

In our training process, we implement a tailored loss function that combines data loss and physics loss components. The data loss ($\mathcal{L}$) is calculated as the Mean Square Error (MSE) between the fine mesh data and the predicted fine mesh data. The physics loss, on the other hand, is determined by computing the MSE of the residual of the governing equations between the fine mesh and the predicted mesh data. The total loss function is expressed as:

Total Loss Function: $$\mathcal{L}_{total} = \mathcal{L}_{data} + \mathcal{L}_{physics} \tag{7}$$

We have tailored the weighted physics loss for our dataset to capture the relevant physical constraints. Adjustments to this loss expression can be made to adapt to different physics scenarios within the dataset. The physics losses for the different datasets are defined here.

Physics Loss: $$\mathcal{L}_{physics} = \alpha_{conv}\mathcal{L}_{conv} + \alpha_{diff}\mathcal{L}_{diff} \tag{8}$$

2D-Burger Equation We have set the weight for the convective term α_{conv} is $1e^{-4}$ and the weight of the diffusive term α_{diff} is $5e^{-10}$.The order of these weights is relatively lower because during our training process, we noticed that the convective term and diffusive term had orders of magnitude around e^4 and e^{10}, respectively, which were higher than the data loss component. In order to maintain a consistent order of magnitude across all components, we employed a trial-and-error approach to determine these weight values. This same rationale applies to our treatment of other datasets.

$$\begin{aligned}
\mathcal{L}_{conv} &= |(\mathbf{u}\cdot\nabla\mathbf{u})_{prediction} - (\mathbf{u}\cdot\nabla\mathbf{u})_{fine}| \\
\mathcal{L}_{diff} &= |(\nabla^2\mathbf{u})_{pred} - (\nabla^2\mathbf{u})_{fine}|
\end{aligned} \tag{9}$$

2D-Methane Combustion we have set the weight for the convective term α_{conv} is $1e^{-5}$ and the weight of the diffusive term α_{diff} is $1e^{-9}$.The value of weight α_{conv} and α_{diff} are obtained by trial and error method.

$$\begin{aligned}
&\text{convective term} = \mathbf{u}\cdot\nabla\mathbf{u} + \nabla\cdot(\mathbf{u}h + \mathbf{u}\frac{u^2}{2}) + \nabla\cdot\mathbf{u}Y_i \\
&\mathcal{L}_{conv} = |(\text{convective term})_{pred} - (\text{convective term})_{fine}| \\
&\mathcal{L}_{diff} = |(\nabla^2\mathbf{u} + \nabla^2 h + \nabla^2 Y_i)_{pred} - (\nabla^2\mathbf{u} + \nabla^2 h + \nabla^2 Y_i)_{fine}|
\end{aligned} \tag{10}$$

Industrial Heat Exchanger we have set the weight for the convective term α_{conv} is $1e^{-6}$ and the weight of the conductive term α_{cond} is $1e^{-10}$.The value of weight α_{conv} and α_{cond} are obtained by trial and error method.

Physics Loss:
$$\mathcal{L}_{physics} = \alpha_{conv}(\mathcal{L}_{conv,m} + \mathcal{L}_{conv,f}) + \alpha_{cond}\mathcal{L}_{cond} \tag{11}$$

$$\begin{aligned} \mathcal{L}_{conv,m} &= |(\partial_{\varphi}T_m)_{prediction} - (\partial_{\varphi}T_m)_{fine}| \\ \mathcal{L}_{conv,f} &= |(\partial_z T)_{prediction} - (\partial_z T)_{fine}| \\ \mathcal{L}_{cond} &= \left|(\partial_z^2 T_m)_{prediction} - (\partial_z^2 T_m)_{fine}\right| \end{aligned} \tag{12}$$

The derivatives of the governing equations in the physics loss were computed using a 2nd-order finite difference method (FDM). Although there is an intention to explore higher-order FDM techniques in the future for potentially improved resolution in data feature prediction.

This loss function formulation allows for the simultaneous optimization of data fidelity and adherence to physics constraints during the training of the model.

5.5 Comparative Analysis: Downsampling vs. Coarse Mesh Representation

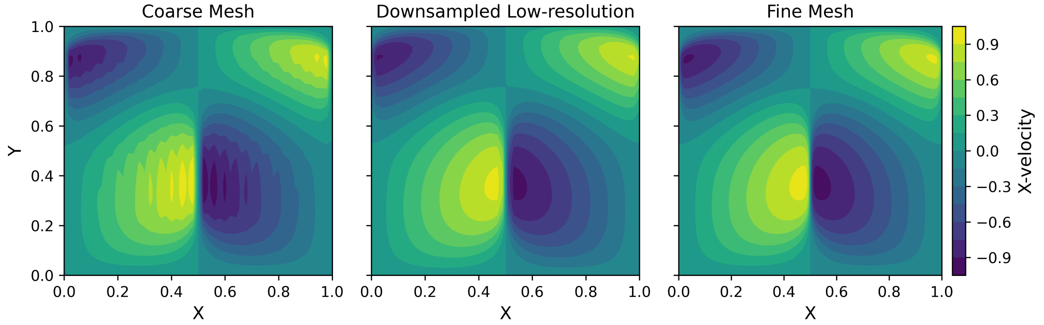


Figure 5: Comparison of Contour Surface Plots for U_x in 2D-Burgers eq: Coarse Mesh vs. Downsampling vs. Fine Mesh Data

Figure 5 demonstrates that generating low-resolution data by simply downsampling the high-resolution data retains the majority of the underlying physics. In this context, we applied average pooling to downsample the fine-mesh data (high-resolution) by a factor of 8, resulting in a low-resolution grid of 51x51, which matches the resolution of our corresponding coarse-grid data. Additionally, figure 6, which examines the convective term ($\mathbf{u} \cdot \nabla \mathbf{u}$), further substantiates our claim. To facilitate a fair comparison, we also applied the vanilla UNet super-resolution technique to both the downsampled low-resolution data and the coarse mesh data. From the results presented in Table 1, it is evident that vanilla UNet is more successful at super-resolving the downsampled low-resolution

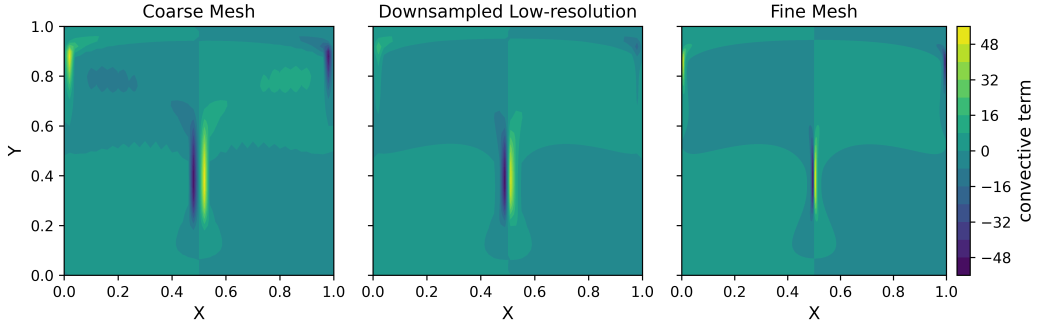


Figure 6: Comparison of Contour Surface Plots for $\mathbf{u} \cdot \nabla \mathbf{u}$ in 2D-Burgers eq: Coarse Mesh vs. Downsampling vs. Fine Mesh Data

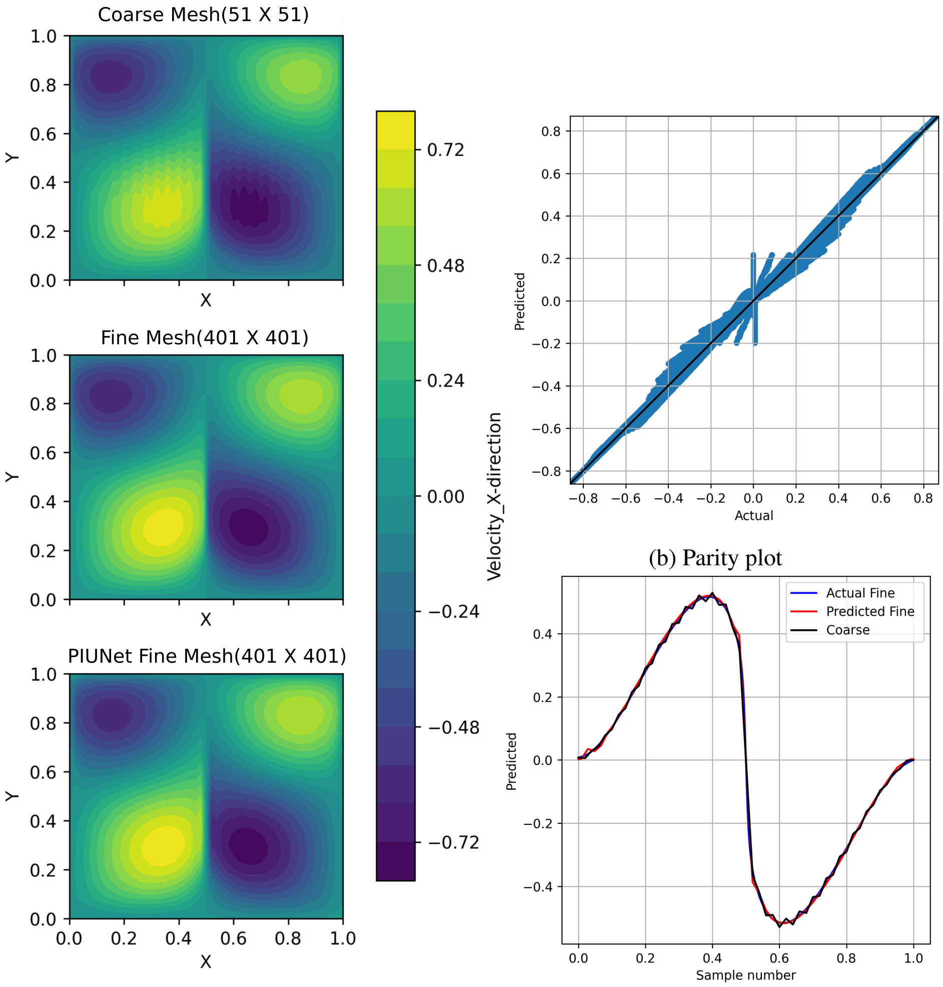
Table 3: Comparison of vanilla UNet performance on downsampled low-resolution data and coarse mesh data.

Algorithm	Data	RMSE	MAE	R^2
UNet	Downsampled low-resolution (51 × 51)	0.02125	0.00426	0.9944
	Coarse Mesh (51 × 51)	0.0283	0.01265	0.9955
PIUNet	Coarse Mesh (51 × 51)	0.02075	0.00628	0.9948

data to the fine mesh level compared to the coarse mesh data. This outcome is attributed to the fact that the downsampled low-resolution data retains a higher degree of the governing physics from the high-resolution data. Remarkably, the UNet super-resolution result of the downsampled low-resolution data is on par with that of PIUNet.

5.6 Additional plots

2D-Burger Equation



(b) Parity plot

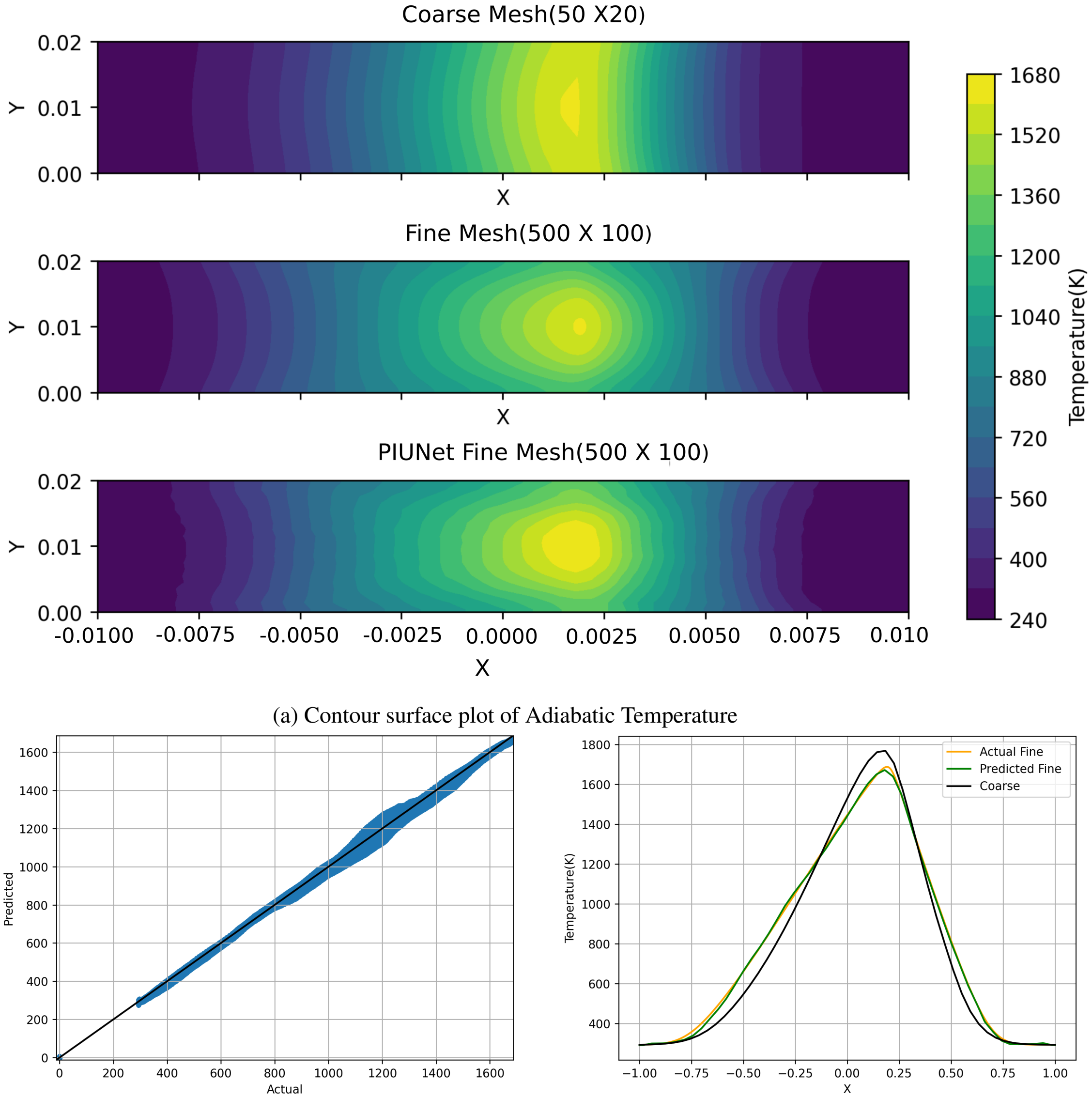
(a) Contour Surface Plot of x-velocity

(c) Centerline x-velocity trend plot

Figure 7: Velocity plots of 2D-Burger Equation

In Figure 7, the contour surface plot provides a visual representation of PIUNet's proficiency in predicting the solution for the fine mesh grid (401×401) with a mean L2 error of 0.0097 when compared to the actual fine mesh data. Conversely, for the coarse mesh grid (51×51), the mean L2 error is 0.0446 in comparison to the actual fine mesh data, showcasing a significant reduction of approximately 78% in the mean L2 error achieved through PIUNet's predictions. This reduction underscores the effectiveness of PIUNet in enhancing accuracy. Furthermore, upon examining the centerline trend plot of U_x, it becomes evident that PIUNet adeptly replicates the observed behavior present in the actual fine mesh data. However, it's worth noting that, particularly in the vicinity of the discontinuity region, there are some values with small mismatches between the actual and predicted data, as observed in the parity plot.

Methane Combustion



(a) Contour surface plot of Adiabatic Temperature

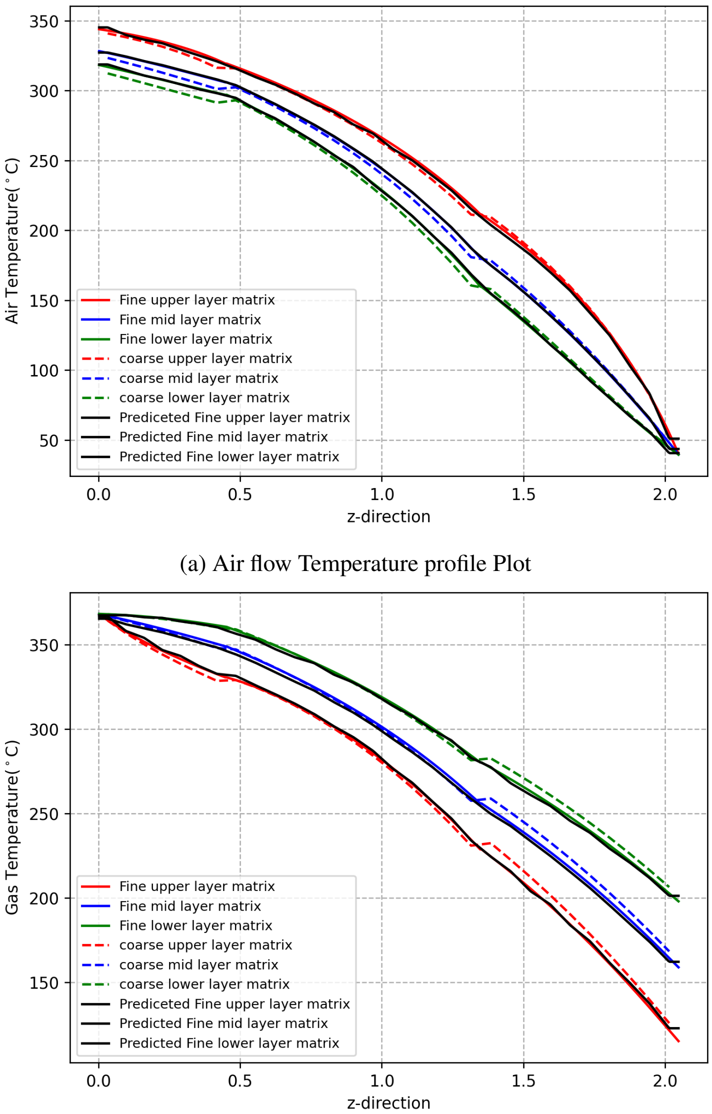
(b) Parity plot

(c) Centerline Temperature trend plot

Figure 8: Temperature plots of Methane combustion

In the case of the methane combustion experiment, the adiabatic flame temperature holds significant importance. The contour surface plot, depicted in Figure 8, clearly demonstrates that PIUNet excels in capturing the gradient of each counter in the fine mesh simulation from coarse mesh data. Furthermore, the parity plot and the centerline temperature profile reveal that PIUNet accurately predicts the entire adiabatic flame temperature profile, including the peak temperature value.

Industrial Heat Exchanger



(a) Air flow Temperature profile Plot

(b) Gas flow Temperature profile Plot

Figure 9: Temperature profile comparison of coarse, fine, and predicted fine mesh from PIUNet for Industrial Heat Exchanger