

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Generative AI as surrogate models for superresolution fluid flow simulations

José Leandro Rodrigues da Silva



Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Zafeiris Kokkinogenis

Second Supervisor: Mónica Silva Ph.D

July 31, 2025

Generative AI as surrogate models for superresolution fluid flow simulations

José Leandro Rodrigues da Silva

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Carlos Soares

Referee: Prof. Francesco Renna

Referee: Prof. Zafeiris Kokkinogennis

July 31, 2025

Resumo

A Dinâmica de Flúidos Computacional é uma técnica essencial para diversas indústrias e domínios de investigação, permitindo a análise e estudo do comportamento de flúidos em diversas condições. As aplicações desta ferramenta incluem engenharia aeroespacial, meteorologia e biomedicina. A principal limitação da técnica de dinâmica de flúidos computacional prende-se com o elevado custo de produção de simulações de elevada fidelidade, em termos de tempo e de recursos computacionais, limitando a sua acessibilidade e democratização. Avanços recentes em Inteligência Artificial e métodos baseados em dados têm demonstrado um grande potencial para técnicas de Super-Resolução, que têm como objetivo melhorar simulações de baixa resolução, transformando-as em simulações de alta resolução, reduzindo drasticamente os custos computacionais associados, mantendo níveis elevados de precisão.

Este trabalho explora o uso de modelos generativos de inteligência artificial, modelos de difusão em particular, como modelos substitutos para tarefas de SR em simulações de fluxo de fluidos. É conduzida uma investigação sobre como esses modelos podem ser usados para refinar os resultados dos métodos numéricos tradicionais, em termos de precisão e eficiência, com foco na incorporação de restrições físicas para garantir resultados fisicamente consistentes e realistas. Uma grande lacuna na literatura é a falta de métricas de avaliação confiáveis para avaliar tanto a fidelidade física como a precisão percetiva. Ao integrar restrições físicas inovadoras e desenvolver uma estrutura de avaliação robusta, este trabalho visa melhorar a generalização, interpretabilidade e confiabilidade de modelos generativos no contexto de simulações de fluidos.

A metodologia proposta inclui a seleção de modelos generativos de última geração, a triagem de diversos conjuntos de dados e a realização de diversas experiências que permitam uma análise comparativa e a validação de modelos. Ao identificar sistematicamente as limitações dos modelos, testar adaptações e implementar condicionamento físico, este estudo refinará abordagens existentes e contribuirá com novos *insights* para a área. Os resultados esperados incluem uma plataforma para a realização de várias experiências validada para avaliar modelos generativos em aplicações de dinâmica de flúidos computacional, estratégias aprimoradas para condicionar modelos de IA com restrições físicas e uma análise do potencial destes modelos para democratizar simulações de alta fidelidade, tornando-as mais viáveis computacionalmente.

Através desta investigação, pretende-se responder as seguintes questões de investigação: Modelos de difusão podem, efetivamente, melhorar simulações de baixa resolução? Como condicionar modelos generativos através de leis e/ou aspetos físicos para garantir que tarefas de super-resolução em dados de simulações de flúidos produzam resultados realísticos e fisicamente realizáveis? Quais são os desafios e diferenças de utilizar, como dados de entrada, simulações artificialmente degradadas e simulações, mais realistas, geradas com malhas de menor resolução? A introdução da métrica Gradient Mean Squared error na função de perda permite refinar as previsões dos modelos? É possível obter resultados mais rapidamente ou com melhor qualidade aplicando técnicas de transfer learning, neste contexto?

A experiências realizadas demonstram que os modelos de difusão são eficazes em tarefas de

super-resolução de dados obtidos através de dinâmica de fluídos computacional degradados significativamente, alcançando baixas taxas de erro e reconstruções visualmente precisas. No entanto, o seu desempenho foi limitado em dados obtidos através de simulações de baixa resolução reais, onde os inputs já estão próximos das simulações de alta resolução, mostrando melhorias mínimas. Estratégias avançadas de condicionamento físico dos modelos superaram abordagens mais simples, mas nenhuma superou a baseline nos dados de simulações de baixa resolução reais. A incorporação de métricas como GMSE, quando combinado com o MSE, melhorou a fidelidade dos gradientes sem comprometer a precisão global, enquanto o transfer learning ajudou a reduzir as métricas de erro em comparação com os seus equivalentes sem transferência, embora os resultados qualitativos ainda mostrassem degradação. Estes resultados evidenciam tanto o potencial como as limitações dos modelos de difusão em tarefas de super-resolução em dinâmica de fluídos computacional e sublinham a importância de dados, condicionamento e estratégias de avaliação adequados.

Abstract

Computational Fluid Dynamics (CFD) is an essential technique across various industries and research domains, enabling the analysis and study of fluid behavior under different conditions. Applications of this tool include aerospace engineering, meteorology, and biomedicine. The main limitation of CFD lies in the high cost of producing high-fidelity simulations, both in terms of time and computational resources, which restricts its accessibility and broader adoption. Recent advances in Artificial Intelligence (AI) and data-driven methods have shown significant potential for Super-Resolution (SR) techniques, which aim to enhance low-resolution (LR) simulations into high-resolution (HR) ones, drastically reducing computational costs while maintaining high levels of accuracy.

This work explores the use of generative AI models, specifically, diffusion models (DMs), as surrogate models for SR tasks in fluid flow simulations. It investigates how these models can be used to refine the results of traditional numerical methods in terms of accuracy and efficiency, with a focus on incorporating physical constraints to ensure realistic and physically consistent outcomes. A major gap in the literature is the lack of reliable evaluation metrics that can assess both physical fidelity and perceptual accuracy. By integrating innovative physical constraints and developing a robust evaluation framework, this work aims to improve the generalization, interpretability, and reliability of generative models in the context of fluid simulations.

The proposed methodology includes the selection of state-of-the-art generative models, screening of diverse datasets, and construction of a unified platform to facilitate comparative analysis and model validation. By systematically identifying model limitations, testing adaptations, and implementing physical conditioning, this study will refine existing approaches and contribute new insights to the field. The expected outcomes include a validated framework for assessing generative models in CFD applications, improved strategies for conditioning AI models with physical constraints, and an analysis of the potential of these models to democratize high-fidelity simulations by making them more computationally feasible.

Through this investigation, the aim is to address the following research questions: Can DMs be used to accurately upsample AifRANS simulations? How to condition generative models with physical laws and parameters to ensure that SR tasks on fluid flow simulation data produce realistic results that really represent the phenomena under study? What are the challenges and the differences between using artificial downsampled HR simulations as LR and real LR simulations? Does GMSE in the loss function improve CFD predictions? Can transfer learning help DMs in the context of CFD simulations achieve better or faster results?

The experiments demonstrate that DMs are effective at super-resolving heavily downsampled, synthetic CFD data, achieving low error rates and visually accurate reconstructions. However, their performance was limited on real LR data, where inputs are already close to HR, showing minimal improvements. Advanced physical conditioning strategies outperformed simpler approaches, but none surpassed the baseline on real LR data. GMSE, when combined with MSE, improved gradient fidelity without sacrificing global accuracy, while transfer learning helped re-

duce error metrics compared to their non-transfer counterparts, although qualitative results still showed degradation. These findings highlight both the potential and limitations of DMs in CFD SR tasks and underscore the importance of appropriate data, conditioning, and evaluation strategies.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a better and more sustainable future for all. It includes 17 goals to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

This dissertation directly supports the SDGs by advancing computational techniques that can improve the efficiency, accessibility, and environmental sustainability of complex simulations. The development of surrogate models for fluid flow simulations has a dual purpose: reducing the computational load of scientific modeling, improving the performance of simulation tools, and aiding in making these kinds of tools more accessible and manageable to a wider range of users and applications.

By enabling faster and more accurate modeling techniques that can be applied to vast important domains, including environmental systems, such as ocean currents, weather patterns, and industrial fluid dynamics, this research supports improved decision-making in climate policy, infrastructure planning, and resource optimization. Additionally, the models and methods developed in this research have the potential to be shared through future open-access publications, academic collaboration, and training events, thereby contributing to broader knowledge dissemination and capacity building.

The relevance of this research to sustainable development is most closely aligned with two SDGs:

SDG 9: Industry, Innovation, and Infrastructure — Encourages development of quality, reliable, sustainable infrastructure and fosters innovation.

SDG 13: Climate Action — Calls for urgent action to combat climate change and its impacts.

SGD	Target	Contribution	Performance Indicators and Metrics
9	9.5	Enhance scientific research and upgrade the technological capabilities of industrial sectors.	Reduction in simulation runtime, improved resolution support industrial innovation, and strengthened knowledge on SR tasks for CFD.
13	13.3	Improve education, awareness-raising, and human and institutional capacity on climate change mitigation and adaptation..	More efficient training reduces computational costs and environmental impact, contributing to sustainable and climate-conscious research and development.

Acknowledgements

Firstly, I want to thank my supervisors, Prof. Zafeiris Kokkinogenis and Prof. Mónica Silva, for all the guidance, readiness, attentiveness, and support during this period. Every valuable insight and criticism was deeply appreciated. Having two expert faculty members in topics that genuinely interest me is something I am undoubtedly thankful for.

For all the unwavering support, unconditional love, and encouragement, I am profoundly grateful to my mom, dad, and younger sister. I am certain that, anytime, especially in the most difficult times, they are always there for me. Whether I need to get something off my chest or simply unwind after a long day, I never doubt their ability to provide me with all the comfort, understanding, and space I need.

Also, I would like to express my appreciation to all my friends, who definitely made this journey way more enjoyable. I could not be more thankful for all the years of providing me with amazing memories, fun moments, and companionship.

Leandro Silva

“Shoot for the stars but if you happen to miss, shoot for the moon instead”

Neil Armstrong

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Problem	3
1.4	Research questions	5
1.5	Document Structure	5
2	Background	7
2.1	CFD	7
2.2	Surrogate Modeling	9
2.2.1	Surrogate Modeling Techniques	9
2.2.2	Applications in Engineering and Physics	9
2.3	SR in Image Processing	10
2.4	Reduced Order Modelling (ROM)	10
2.4.1	Proper Orthogonal Decomposition (POD)	11
2.4.2	Autoencoders	12
2.4.3	Galerkin Projection	12
2.4.4	Regression-Based Models	12
2.5	DL	13
2.5.1	RNN	13
2.5.2	CNN	13
2.5.3	GAN	13
2.5.4	DM	14
2.6	Evaluation Metrics	16
2.7	Transfer Learning	17
3	Related Work	18
3.1	Data	18
3.1.1	DAMBREAK	19
3.1.2	Flow Past a Cylinder	19
3.1.3	Smoke	19
3.1.4	Kolmogorov flows	19
3.1.5	Rayleigh-Bénard convection	19
3.1.6	Aerodynamics	20
3.1.7	Bubbly flow	20
3.1.8	SuperBench	20
3.1.9	MegaFlow2D	21
3.1.10	Discussion	21

3.2	Innovative generative models and strategies for fluid flow predictions	22
3.2.1	CNN-based strategies	22
3.2.2	GAN-based strategies	23
3.2.3	Diffusion-based strategies	23
3.2.4	Other strategies	25
3.2.5	Discussion	25
3.3	Physics-informed Strategies	26
3.3.1	Discussion	27
3.4	Metrics for evaluating generated images	28
3.4.1	Pixel-level metrics	28
3.4.2	Perceptual metrics	28
3.4.3	Flow field-specific metrics	29
3.4.4	Discussion	30
3.5	Transfer Learning	30
3.5.1	Discussion	32
3.6	Final remarks of literature review	32
4	Data & Methods	34
4.1	Data	34
4.1.1	Pre-processing	35
4.1.2	Initial conditions	35
4.1.3	LR simulation data	35
4.1.4	Data extraction for transfer learning experiments	36
4.1.5	Summary and directory structure	36
4.2	Methodology	37
4.2.1	Research foundation	37
4.2.2	Core Idea	41
4.2.3	Architectural and Functional Restructuring	42
4.2.4	Configuration	46
4.2.5	Runner	46
4.2.6	Training	46
4.2.7	Validation	47
4.2.8	Testing	47
4.2.9	Logging	47
5	Experiments	48
5.1	Overview of Experimental Goals	48
5.2	Baseline experiments	49
5.3	Physical conditioning	50
5.4	Dataset comparison	52
5.5	Real LR Data	53
5.6	Single-channel input	55
5.7	GMSE	57
5.8	Advanced Physical conditionings	59
5.9	Transfer Learning	61
5.10	Discussion	63

6 Conclusion	66
6.1 Main Contributions	66
6.2 Futher Developments	69
References	71

List of Figures

2.1	Schematic of ROM methods, including POD, AE, Galerkin projection, and regression models, adapted from [66].	11
2.2	Reconstruction of velocity components u , v , and pressure p using different POD mode combinations for flow around a cylinder, adapted from [52].	11
2.3	Autoencoder architecture, adapted from [35]	12
2.4	CNN architecture, adapted from [43]	14
2.5	Backbone architecture of a DDPM: U-net architecture, adapted from [54]	16
3.1	SuperBench samples from three different scientific domains, adapted from [51] .	21
4.1	Visualization of different target variables from the AirfRANS dataset using PyVista. Each image corresponds to a scalar field plotted over the same airfoil mesh. . . .	35
4.2	Comparison between RGB and single-channel real LR airfoil simulations.	36
4.3	Barati workflow illustration, adapted from [10]	40
4.4	Example of logging during a training stage	47
5.1	8× Upsampling without Physical Conditioning.	51
5.2	Loss evolution curves during training for the baseline experiments. Each curve shows the training loss as a function of training steps for the different configurations.	52
5.3	8× Upsampling with Timestep Embedding (MSE + SSIM).	53
5.4	Loss evolution during training for the physical conditioning experiments.	54
5.5	8× Upsampling with real LR input data.	55
5.6	Loss evolution curves during training for real LR data experiments.	56
5.7	8× unconditioned upsampling with single-channel input.	57
5.8	Loss evolution curves during training for single-channel input experiments. . . .	58
5.9	8× unconditioned upsampling with GMSE-only loss function.	60
5.10	Loss evolution curves during training for the GMSE experiments.	61
5.11	Visualization example of a <code>exp_imp</code> result.	62
5.12	Visualization example of a <code>exp_imp_full</code> result.	63
5.13	Loss evolution curves during training for the advanced physical conditioning experiments.	64
5.14	Visualization example of a <code>exp_imp</code> result with transfer learning.	65
5.15	Visualization example of a <code>exp_imp_full</code> result with transfer learning.	65
5.16	Loss evolution curves during training for the advanced conditioning strategies with transfer learning.	65

List of Tables

5.1	Comparison of different upsampling levels using MSE loss only	49
5.2	Comparison of different upsampling levels using MSE + SSIM loss	50
5.3	Comparison of physical conditioning methods using MSE loss	50
5.4	Comparison of physical conditioning methods using MSE + SSIM loss	50
5.5	Comparison between LR datasets and their HR counterparts	53
5.6	Comparison of strategies using real LR data	55
5.7	Comparison of single-channel (grayscale) strategies	57
5.8	Comparison of strategies using GMSE in the loss function	59
5.9	Comparison of advanced physical conditioning strategies	60
5.10	Comparison of experiments applying transfer learning	62

List of Acronyms

AE	Auto-Encoder
AI	Artificial Intelligence
ANN	Artificial Neural Networks
AoA	Angle of Attack
BAPPS	Berkeley-Adobe Perceptual Patch Similarity
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
CV	Computer Vision
daGAN	Data Augmented Generative Adversarial Network
DL	Deep Learning
DM	Diffusion Model
DNS	Direct Numerical Simulation
EMA	Exponential Moving Average
FEA-PSNR	Free Energy Adjusted Peak Signal-to-Noise Ratio
GAN	Generative Adversarial Network
GMSE	Gradient Mean Squared Error
HR	High Resolution
KAN	Kolmogorov–Arnold Network
LDM	Latent Diffusion Models
LES	Large-Eddy Simulation
LPIPS	Learned Perceptual Image Patch Similarity
LR	Low Resolution
LSTM	Long-Short Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multi Layer Perceptron
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
NACA	National Advisory Committee for Aeronautics
NIQE	Natural Image Quality Evaluator
PCA	Principal Components Analysis
PDF	Probability Density Function
PIML	Physics Informed Machine Learning
PINN	Physics-Informed Neural Network
PIV	Particle Image Velocimetry
POD	Proper Orthogonal Decomposition
PSNR	Peak Signal-to-Noise Ratio
RANS	Reynolds-Averaged Navier-Stokes
RBF	Radial Basis Function

RFNE	Relative Frobenius Norm Error
RNN	Recurrent Neural Networks
ROM	Reduced Order Modelling
SDE	Stochastic Differential Equations
SISR	Single-Image Super-Resolution
SR	Super-Resolution
SSIM	Structural Similarity Index
SVR	Support Vector Regression
TKE	Turbulent Kinetic Energy
UDM	Uncertainty Quantification
VOF	Volume of Fluid

Chapter 1

Introduction

This first chapter introduces the foundational aspects of the work. It outlines the context in which the research is situated, presents the motivation for exploring this particular technological domain, and defines the core problem addressed. It also formulates the research questions that guide the investigation, framed in an open-ended and exploratory manner. Finally, the chapter concludes with a description of the overall structure of the document.

1.1 Context

Analyzing the behavior of fluids, whether they are liquids or gases, is immensely helpful in multiple industry sectors, as evidenced in [9, 28, 39, 42, 65], to understand the impacts of fluid dynamics on a system's performance, safety, efficiency, and viability. Since fluids make up a large portion of matter on Earth, their behavior plays a fundamental role in shaping both natural phenomena and industrial processes. Companies and engineers heavily depend on experimental and numerical methods to study and understand either industrial or natural phenomena. Fluid simulations offer significant advantages for testing prototypes and predicting interactions, making them essential tools. However, understanding the interaction of fluids is still a highly complex challenge, and detailed numerical simulations are one of the strategies adopted.

To illustrate the benefits of CFD, an aircraft manufacturing company needs to simulate fluids to test how newly designed airfoils will perform under multiple flight environments and conditions. CFD, as presented in [6, 27], helps the company validate designs and predict behavior. Another example is in atmospheric studies, such as [4, 65, 79], where CFD is used to indicate the movement of weather systems, analyze pollutant dispersion, and model wind patterns for renewable energy applications, such as optimizing the placement of wind turbines, as studied in [7, 75].

CFD uses various numerical techniques and modeling strategies designed to solve the Navier-Stokes equations and other governing equations for fluid flows. These methods vary in fidelity and computational cost, ranging from simplified models like Reynolds-Averaged Navier-Stokes (RANS) to more detailed approaches such as Large Eddy Simulation (LES) and Direct Numerical

Simulation (DNS). DNS, in particular, produces very high-fidelity results, but due to the high computational costs, it is not widely used in industry.

The central challenge and complexity with modeling/simulating fluids comes from the need to solve the complex equations, such as the Navier-Stokes equations. These equations dictate fluid dynamics through mass, momentum, and energy conservation principles. Being partial differential equations (PDEs), the Navier-Stokes equations can exhibit complex, time-dependent, and chaotic behavior, especially in turbulent flows [67]. Due to their sensitivity to initial conditions and the need to resolve multiple interacting scales, solving these equations typically requires sophisticated numerical methods.

DNS is a high-fidelity CFD approach that resolves all flow structures without turbulence modeling. In many cases, it is computationally intensive and hard to control for most real-world applications [67]. To reduce the computational cost of CFD, low-resolution simulations are often used as a compromise, achieved by coarsening the mesh or simplifying equations. SR is a promising technique to mitigate this compromise by reconstructing high-fidelity results from simplified inputs.

SR is a task that involves recovering fine-scale data from corresponding coarse-grained data [51], so it's a technique that can assist CFD simulations by, essentially, upscaling LR simulations to HR ones. If such a technique could be successfully applied in this context, an expensive HR simulation would not be needed to obtain high-fidelity data. SR will then lead to a significant reduction in time and resources.

1.2 Motivation

CFD simulations are used as a method for solving physical problems numerically, leading to high volumes of data, requiring detailed data, and consuming many computational resources. The required data to simulate fluids usually involves a definition of the domain, boundary conditions, and the initial conditions. Each component can be a set of parameters with a wide range of values, which may lead to a high number of simulations being a time-consuming as well as a resource-consuming process. Having a tool that can efficiently produce high-quality CFD simulations from low-quality ones would be a powerful resource and should retain as much detail as possible while achieving a high level of physical accuracy. Having access to reliable, fast, and on-demand CFD simulations would be a game changer for multiple industries and research fields, including meteorology, healthcare, aerospace engineering, land/aerospace/marine vessel design, astronomy, and oceanography. The impact stems from the fundamental importance of fluid dynamics in these areas, where accurate and timely flow analysis is essential for forecasting, design optimization, safety assessments, and the study of complex physical systems.

As previously discussed, the use of SR represents a potential breakthrough in this context and is currently an active area of research, primarily driven by advancements in AI and Machine Learning (ML) techniques. With Deep Learning (DL) models being the standard approach to tackle SR, these intelligent methods have also seen major advancements in this field, and have

achieved very promising early results, as evidenced by [12, 60, 73]. These technologies can essentially move the complexity of inferring results to a training phase, and by learning patterns from current data, they make it possible to make reliable predictions on future data.

Surrogate models are simplified computational models that approximate complex simulations or processes, allowing for faster and more efficient analysis while retaining essential details. Generative AI can therefore be applied in SR in the context of CFD simulations as surrogate models. This specific type of AI tries to directly generate the full high-fidelity domain.

With this in mind, we can establish the motivation for researching these technologies in this context. DL strategies include Generative Adversarial Networks (GAN) and DMs, but could also include Convolutional Neural Networks (CNN), Auto-Encoders (AE), Recurrent Neural Networks (RNN), and many others. These models are being extensively studied in multiple fields and have shown high potential at the task of upsampling LR inputs with different data domains; examples are shown in [29, 58].

An example of a solution can be found in [51], where a novel model, called the Combined Model, composed of a conditional denoising DM and an enhanced residual network (EResNet), was developed to reconstruct high-resolution (HR) turbulent flow fields from LR flow data. The results attained with the Combined Model exhibit high conformity with the DNS results in terms of flow distribution, details, and accuracy.

Generative AI technologies have the potential to significantly benefit industries reliant on fluid dynamics understanding through CFD simulations, making it accessible to smaller companies and researchers, especially those with limited computational resources for high-quality simulations. This reduction of the barriers to access these technologies can democratize research and enable innovative use cases that were previously unattainable due to the time and cost constraints of detailed simulations.

1.3 Problem

The computational intensity and time demands of high-fidelity CFD simulations present a significant barrier to their widespread application. While these simulations are essential for understanding and optimizing fluid dynamics, the resources required to produce high-quality results often exceed the capabilities of smaller organizations or research groups. As mentioned, Generative AI coupled with SR offers a promising approach to address this issue by enhancing LR simulations to high-resolution outputs.

However, several challenges limit the effective application of data-driven models in CFD. Notable suggestions and identified limitations of ML solutions for this type of problem are outlined in [51] and [67]. [51] mentions, as challenges, (1) validation, (2) benchmarks on generalization, (3) Uncertainty quantification (UQ), (4) dimensionalization problem, which involves understanding the correlation between the network's architecture, deepness, structure, and size, and the physical parameters, and (5) the design of an evaluation metric that quantifies the solution quality. In addition to these challenges, the work in [67] also advocates for the importance of identifying

areas where ML outperforms classical methods because such models may be more accurate and efficient. The authors also recognize the possible benefits that transfer learning techniques may bring to the question of the associated cost of benchmarking and how the data is generated.

CFD simulations, and fluid dynamics as a whole, are a very complex physics problem that involves many variables and frequently requires approximations. DL models are expensive to train and need a lot of data for their learning process. A simple solution could be to train these models on only the LR image plots generated by numerical simulations. The plots allow the visualization of many physical properties of the fluid being studied, whether it's pressure, velocity, temperature, density, or vorticity. However, when working with complex physical systems, this solution can be insufficient.

AI/ML models, when faced with this kind of problem, benefit substantially from incorporating physics Laws directly in their learning process. For the problem at hand, the Navier–Stokes equations and any other kinematic or thermodynamic constraints can be directly incorporated in the loss function of the neural network by penalizing deviations from the target values and are properly weighted with any given data [10].

Neural Networks constrained to physical equations and/or constraints are called Physics-Informed Neural Networks (PINNs). This kind of network actually accelerates neural network training, improves model generalization with less data, and manages complex applications while ensuring solutions adhere to physical laws [5]. From these networks, we get physically accurate results that take advantage of not just the image data but also the detailed flow variables associated with each component of the simulation. However, PINNs have some limitations, as identified in [10], including unstable training due to a highly non-convex loss landscape, lower accuracy compared to traditional CFD solvers for forward problems, and strong sensitivity to data quality and loss term balancing.

The main focus of this research is studying how CFD simulations can become more accessible by making them easier to compute through the use of surrogate Generative AI models for SR tasks on fluid flow simulations. There are many paths to and possible solutions to faithfully predict and/or enhance fluid flow simulations. The outcome of this work should be a deep research and set of experiments that build upon the current state-of-the-art by leveraging innovative strategies, physical restraints, and an evaluation system allowing the identification of solid generative models with accurate and efficient results.

To support this investigation, the airfRANS dataset, presented in [8], is used as a benchmark for SR in CFD. It provides HR RANS aerodynamic simulations across varying airfoil shapes and flow conditions, making it ideal for training and evaluating models in contexts such as airfoil analysis and optimization. Moreover, since traditional pixel-level metrics, such as Mean Squared Error (MSE), often fail to capture physically meaningful errors, this work also employs the more advanced Gradient Mean Squared Error (GMSE) [15], which explicitly accounts for the gradients in the predicted fields and provides a more physically realistic assessment.

1.4 Research questions

With the context, motivation, and problem described in the preceding sections, and in order to properly conduct the needed research and study, objectives need to be set. To frame the research needed for this dissertation, the following group of research questions is proposed:

- Can DMs be used to accurately upsample AifRANS simulations?
- How to condition generative models with physical laws and parameters to ensure that SR tasks on fluid flow simulation data produce realistic results that really represent the phenomena under study?
- What are the challenges and the differences between using artificial downsampled HR simulations as LR and real LR simulations?
- Does GMSE in the loss function improve CFD predictions?
- Can transfer learning help DMs in the context of CFD simulations achieve better or faster results?

1.5 Document Structure

This dissertation is organized into six chapters, each addressing a distinct aspect of the research and development, collectively building toward a comprehensive understanding of the proposed work.

Introduction: Chapter 1 introduces the context and motivation behind the research, clearly defines the problem statement, and presents the overarching research questions that guide the study.

Background: Chapter 2 introduces the key theoretical concepts, foundational concepts, and tools that underpin this research, including CFD principles, ML models, and numerical techniques.

Related works: Chapter 3 performs a complete literature review about the most relevant recent advances in data-driven modeling for fluid dynamics, highlighting relevant approaches, limitations, and gaps in the literature.

Data & Methods: Chapter 4 describes the data sources, tools, and methodological approaches used in the study. This includes a detailed explanation of the research environment, experimental setup, and rationale for methodological choices.

Experiments: Chapter 5 details the experimental procedures and results. It examines the performance and capabilities of DMs in SR tasks, focusing on their effectiveness in fluid dynamics simulations.

Conclusion: Chapter 6 summarizes the main findings and answers the research questions. It also proposes directions for future work, highlighting opportunities for further investigation and practical applications.

This structure ensures a logical progression from foundational concepts to applied experimentation, followed by reflective analysis and contextual relevance, allowing the reader to follow the development of ideas throughout the dissertation.

Chapter 2

Background

This chapter provides the foundational theoretical knowledge required to understand the core ideas explored in this dissertation. It outlines the main concepts and techniques in CFD, numerical simulation, and the role of ML in data-driven modeling. Particular emphasis is given to SR, generative models, and reduced-order modeling—topics that underpin the methodology adopted in this work. By clarifying these essential concepts, this chapter sets the stage for the deeper analysis and innovation presented in the following sections.

2.1 CFD

CFD encompasses techniques where fluid behavior is computationally simulated using the numerical solution of the governing equations for fluid, mainly the NS equations. These equations model conservation of mass and momentum and describe the evolution of variables such as velocity, pressure, and viscosity at specific spatial and temporal instances. One of the most widely used open-source tools for solving CFD problems is OpenFOAM. It provides a comprehensive suite of solvers and utilities for simulating fluid flows and heat transfers.

A key aspect of CFD is the choice of discretization method, described in [80], which converts the continuous governing equations into algebraic forms over a computational domain. The finite volume method is the most widely used due to its conservative formulation and efficiency, especially for large-scale turbulent flows. It ensures mass, momentum, and energy conservation by integrating equations over discrete control volumes. The finite element method, adapted from structural mechanics, offers greater stability and geometric flexibility but at a higher computational cost. The finite difference method is historically significant and simple to implement. It's able to deal with complex geometries accurately. Each method involves trade-offs in accuracy, stability, and efficiency, depending on the problem's complexity and domain geometry.

NS equations, shown in 2.2, are PDEs and therefore, calculating them for turbulent flows can become quite a challenging task. Because turbulent systems have the characteristic of being chaotic and sensitive, small changes in the initial conditions can cause unpredictable behaviors.

$$\text{Continuity : } \nabla \cdot \mathbf{u} = 0 \quad (2.1)$$

$$\text{Momentum : } \rho \frac{d\mathbf{u}}{dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{F} \quad (2.2)$$

These equations describe the behavior of incompressible, Newtonian fluids. The first equation is the continuity equation, which ensures mass conservation by requiring that the divergence of the velocity field $\mathbf{u}$ is zero. The second equation is the momentum equation, expressing the balance of forces acting on a fluid element. In this equation, ρ is the fluid density, and $\frac{d\mathbf{u}}{dt}$ denotes the material derivative of the velocity, capturing both local and convective acceleration. The term $-\nabla p$ represents the pressure gradient force, $\mu \nabla^2 \mathbf{u}$ corresponds to the viscous forces modeled by the Laplacian of the velocity field, where μ is the dynamic viscosity, and $\mathbf{F}$ accounts for external body forces such as gravity.

DNS is the most accurate technique for resolving any fluid flow, because it directly solves the equations numerically, with enough degrees of freedom to resolve all fluid scales, [67]. This fidelity comes at the cost of being, in most cases, computationally unfeasible for many practical problems, especially for flows involving turbulence.

RANS and LES are probably the most well-known techniques for turbulence modelling. Turbulent flows can be abstractly separated into the large flow structures and smaller scales. RANS methods time-average the flow equations, which means that only the mean flow and larger structures are modeled explicitly, while the effects of smaller turbulent fluctuations are ignored. In contrast, LES resolves the large, energy-containing eddies/scales directly, and the smaller, subgrid-scale turbulence is only modelled (approximated), offering improved accuracy at a higher computational cost, just as presented in [67]. There are some attempts and implementations that try to join both methods and produce hybrid methods, such as [34].

Accurate CFD simulations depend not only on the governing equations and discretization method but also on careful attention to boundary conditions, domain discretization, and solver settings. Mesh quality, resolution, and the choice of numerical schemes significantly affect both the stability and accuracy of the results. Poorly defined boundary conditions or overly coarse grids can lead to large errors, particularly in regions with high gradients or turbulence. Additionally, timestep size is important in simulations with temporal evolution, where stability depends on the Courant–Friedrichs–Lewy condition. If not properly set, it can cause the simulation to become unstable or overly damped. CFD models are also sensitive to mesh refinement and topology; insufficient resolution near walls or interfaces can compromise the validity of the solution. As such, setting up a CFD simulation involves balancing numerical stability, physical fidelity, and computational cost through an informed choice of mesh, timestep, and solver configurations.

2.2 Surrogate Modeling

Surrogate modeling aims to provide a simplified and computationally efficient approximation of a complex system or process, emulating its input-output behavior with significantly reduced computational cost, [37]. These models are trained on the input and output structures of the complex system being emulated. This technique is highly useful when applied to high-fidelity CFD simulations, which are too computationally expensive. The main idea is to reduce the complexity of the problem by finding a more efficient path from input to output, potentially trading some accuracy for faster computation and reduced resource requirements, though well-designed models can still achieve high precision in practice.

In [37], some important limitations of surrogate models are outlined: Surrogate models are designed to approximate the exact solution and not to substitute high-fidelity techniques. These kinds of techniques are data-driven and therefore do not have the capacity to interpret the physical reality of the information. This data-driven nature also creates difficulties in extrapolating to data outside the range of data used for training.

2.2.1 Surrogate Modeling Techniques

Surrogate modeling techniques can be broadly categorized into traditional and ML-based approaches. Traditional methods, such as polynomial regression, radial basis functions (RBFs), and Kriging (Gaussian Process Regression), rely on mathematical and statistical approximations to interpolate or approximate sampled input-output data, further described in [37].

ML-based surrogate models such as artificial neural networks (ANNs), CNNs, and support vector regressors (SVRs) offer greater flexibility in capturing complex, nonlinear relationships without explicit assumptions about the underlying physics [37]. These data-driven models have demonstrated strong performance in many applications; however, they generally require larger datasets and careful regularization to avoid overfitting.

2.2.2 Applications in Engineering and Physics

Surrogate models can be used in any problem with a complex system and access to its output and inputs, and have proven effective across various domains. There are reported usages of surrogate models for real-time groundwater flow and contaminant transport analysis [3], in wind energy, they aid in efficient turbine layout optimization by predicting wake effects [71]. For UQ, they replace expensive simulations with probabilistic methods such as Monte Carlo analysis [62]. In civil and electrical engineering, they can help in the task of predicting concrete compressive strength [18] and optimizing electric motor designs [13]. In indoor ventilation, CFD-based surrogates trained on sensor data help estimate flow parameters to improve indoor air quality and energy use [40].

2.3 SR in Image Processing

As already mentioned, SR refers to the process of upsampling the spatial resolution of grid-like information, primarily images. SR typically occurs by transforming an LR input into an HR output. The most naive and traditional SR techniques include interpolation methods such as nearest-neighbor, linear, and cubic upsampling. These are simple but often lead to blurry results, a lack of fine details, and fail to adapt to variations in scale, angle, and lighting conditions, as concluded in [73]. In contrast, modern SR approaches leverage DL models to learn data-driven mappings from LR to HR images, offering significantly better perceptual quality, reconstruction fidelity, and better performance on synthetic datasets, as concluded in [12].

In CFD, SR has become increasingly valuable as a surrogate tool to reconstruct high-fidelity flow fields from coarse simulations or sensor data. This technique retains the flow features and significantly reduces computational cost by approximating the high-fidelity output simulation data from the corresponding LR data obtained from less expensive simulation setups. Multiple and recent works have demonstrated the effectiveness of DL-based SR strategies for enhancing fluid simulation resolution [20, 48, 51, 59, 70], estimating turbulence [30, 84], or predicting time-evolving flow patterns [19, 21, 61]. These applications reflect a growing trend in which SR techniques bridge the gap between expensive HR simulations and real-time or data-limited environments.

2.4 Reduced Order Modelling (ROM)

ROM methods aim to approximate complex fluid dynamics with lower-dimensional representations, significantly reducing computational cost while retaining the most relevant physical features. These types of models are particularly useful in scenarios such as optimization, control, and real-time simulations, where running full high-fidelity CFD simulations would be computationally prohibitive. ROMs typically rely on identifying dominant flow structures and modeling their time evolution. The most relevant and popular ROM methods are presented here, based on the study [66]. An illustrative picture of all the discussed methods is shown in 2.1.

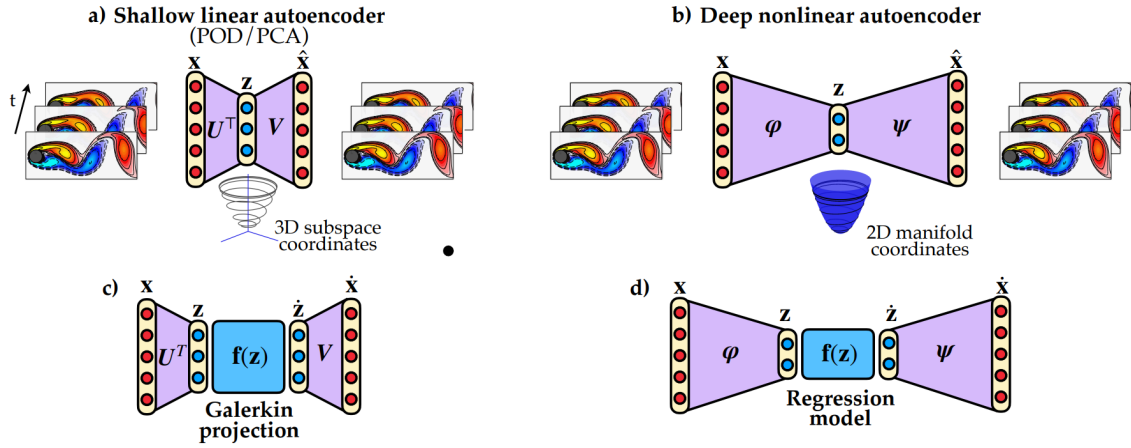


Figure 2.1: Schematic of ROM methods, including POD, AE, Galerkin projection, and regression models, adapted from [66].

2.4.1 Proper Orthogonal Decomposition (POD)

POD is a ROM technique for reducing the dimensionality of fluid flow data by decomposing it into a set of orthogonal spatial modes or eigenfunctions, [52]. The main idea behind POD is to capture the most important characteristics of a specific flow with less data. The obtained modes should capture the most energetic structures in the flow, and with them, it becomes possible to reconstruct a reliable approximation of the initial flow by combining various modes.

In CFD, POD is widely used to accelerate simulations or reconstruct flow fields with fewer degrees of freedom. As shown in Figure 2.2, flow fields reconstructed using only a few leading modes can still capture essential velocity and pressure features, significantly reducing computational cost.

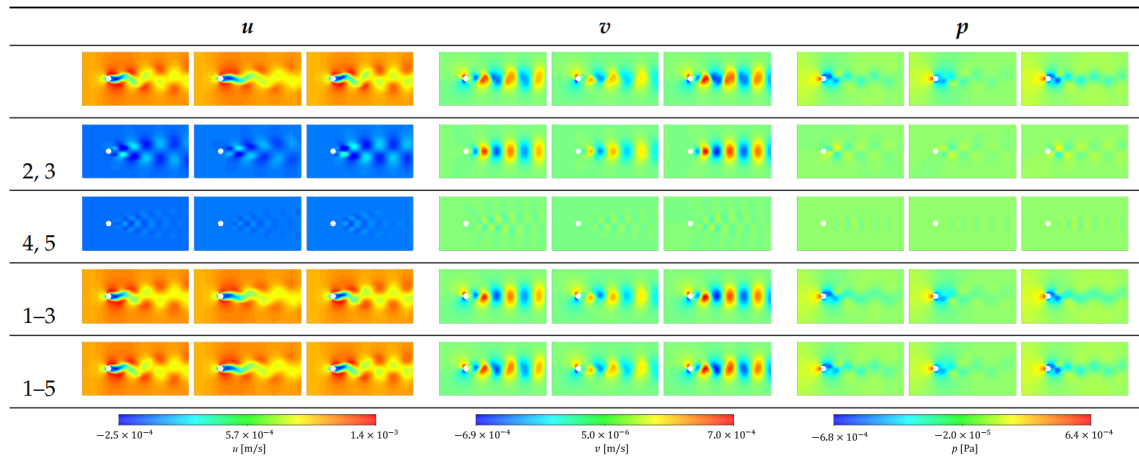


Figure 2.2: Reconstruction of velocity components u , v , and pressure p using different POD mode combinations for flow around a cylinder, adapted from [52].

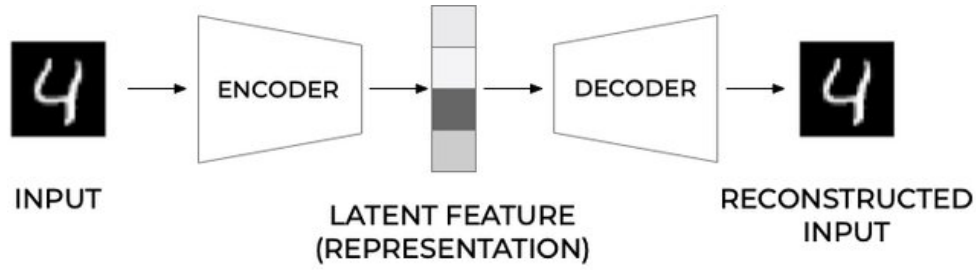


Figure 2.3: Autoencoder architecture, adapted from [35]

2.4.2 Autoencoders

Autoencoders are a type of neural network with the specific purpose of learning a compressed representation of the input data. They are another example of a ROM technique and are able to learn in an unsupervised way; no labeled data is required. The main purpose of this architecture is for dimensionality reduction or extracting the main features of the data, which can be combined to reconstruct any input. The typical architecture of autoencoders, shown in Figure 2.3, includes an encoder and decoder (both neural networks). The encoder compresses the input into a lower-dimensional representation, known as the latent space or bottleneck, which contains the main features of the input data. The decoder's function is to generate/reconstruct the original input from the compressed representation. The study [35] presents and explains in more depth this neural network. A variant of autoencoders, called Variational Autoencoders (VAEs), differs by adding a probabilistic component to the latent space, allowing them to learn distributions rather than fixed encodings. This makes VAEs particularly useful for generating new data similar to the input.

2.4.3 Galerkin Projection

Galerkin projection is a classical approach, described in [66], to obtain reduced-order models by projecting the governing fluid equations onto a reduced basis, often obtained through POD. This results in a set of coupled ordinary differential equations that govern the evolution of the modal amplitudes in time. While effective in capturing flow dynamics, Galerkin models often suffer from stability issues and can require intrusive access to solver internals, making them less flexible in practical applications.

2.4.4 Regression-Based Models

Modern ROM approaches often replace physics-based projections, such as Galerkin, with data-driven regression models, as explored in [66]. Once a low-dimensional representation is established, possibly obtained via POD or autoencoders, various regression techniques, including neural networks, LSTMs, and SINDy, can be used to model the temporal evolution of the reduced coordinates. These methods are typically non-intrusive, making them more compatible with simulation data, and they offer greater flexibility in handling nonlinearities and complex dynamics.

2.5 DL

DL methods are experiencing rapid growth in both research and practical applications, with continuous advancements across a wide range of tasks. In the field of CFD, DL has emerged as a prominent approach for addressing the inherent complexity and high computational cost of fluid simulations, particularly through its use in developing efficient surrogate models.

There are an immense number of different techniques and models, so, in order to give a solid knowledge foundation, only the most commonly used models with CFD are succinctly described.

2.5.1 RNN

RNNs are an interesting type of neural network because they are designed to deal with sequential data, such as time series. The unique feature of RNNs is the ability to maintain a memory of the previous inputs. The previous inputs are then used in combination with the current input to generate the outputs. An evolution of RNNs is LSTM networks, which allow learning the dependencies between data over longer periods of time.

The study [36] presents these concepts and reviews the breadth of applications where RNNs can be used, including NLP tasks such as text generation, Machine Translation, and also Sentiment Analysis, Speech Recognition, Time Series Forecasting, Signal Processing, Bioinformatics, Autonomous Vehicles, and Anomaly Detection.

2.5.2 CNN

CNNs are another type of neural network used extensively in grid-like data, such as images. As shown in Figure 2.4, the typical architecture of CNNs includes convolutional layers, where filter/s/kernels are applied to extract spatial features such as edges, textures, or patterns. After convolutional layers, pooling layers reduce the spatial dimensions of the feature maps (grid) while retaining the essential information, improving efficiency, and reducing the risk of overfitting. Fully connected layers are usually the last component of CNN architectures; they are composed of fully connected neurons that receive the extracted features to make predictions. The study [43] presents this concept in a simplified, yet complete way. These types of operations make CNNs a powerful tool for image pattern recognition.

2.5.3 GAN

GANs were first presented in [22] as generative models that work through an adversarial process. Consists of training two different models, a generator model and a discriminator model. The generator tries to capture the data distribution and generates samples, and the discriminator estimates the probability that a certain sample was generated by the generator or taken from the training data. The goal of GAN training is to maximize the probability of the discriminator making a mistake, therefore achieving a level where the generated samples are very close to the training samples.

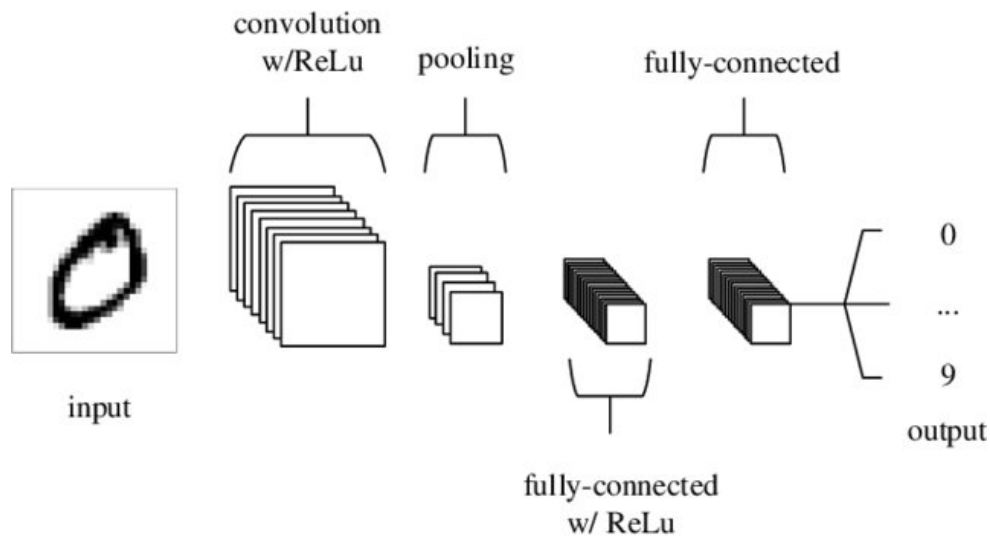


Figure 2.4: CNN architecture, adapted from [43]

2.5.4 DM

DMs are a class of generative models that create new data by iteratively refining a noisy input until a high-quality output is reached.

The workflow of DMs includes a forward process and a reverse process. The forward process gradually adds noise to the input data over several steps until the image is undistinguishable from pure noise; this process follows a certain probabilistic trajectory (Gaussian distribution). The reverse process learns to iteratively remove the noise from noisy data, effectively reconstructing the original data. This process is what is effectively desired for the model to learn.

In the most standardized approach to DMs, represented by the class of Denoising Diffusion Probabilistic Models (DDPMs), both the forward and reverse processes are explicitly modeled as a discrete Markov chain, where each state depends only on the immediate previous state. This memoryless property simplifies training and sampling mathematically, as it allows the model to compute each step based solely on the current state without considering the full history. However, it also introduces a limitation: because the reverse process must be applied step by step through all timesteps, sampling from a DM can be computationally slow compared to other generative models. Despite this, DMs have shown superior performance in producing high-quality, diverse, and stable outputs compared to earlier generative models like GANs, with less risk of mode collapse and better likelihood estimation.

The model only learns the backward process; the forward process happens before the model's inference is mainly used to generate the training samples by adding noise to the samples using a noise scheduler, which also takes into account a specific timestep. The timestep is important because it helps the model to learn how to gradually denoise an image effectively. The resulting images with multiple levels of noise are then passed to a model that uses those images as input to be used in a trained backward process. During training, the predicted images are then compared with the ground-truth samples, and the results are used to improve the model. For the inference,

the model starts with a random noise image and applies the backward process to remove the noise and produce a realistic image. For SR tasks, this differs; the input image is not a fully noisy sample, it's an LR version of the ground-truth image, as described in [59].

To introduce the mathematical principles of DMs, the formulation of DDPMs, presented in [24], will be used as a reference; it is important to clarify some terms. x_0 denotes the original clean data/image, and x_t represents the noisy version of the data at timestep t ; the amount of noise present in the data grows with the timestep. The noise schedule β_t controls the variance of the noise added at each forward step. In the reverse process, the neural network predicts two key parameters at each step: the mean $\mu_\theta(x_t, t)$ of the conditional distribution of x_{t-1} given x_t , and its variance $\Sigma_\theta(x_t, t)$, which may be fixed or learned. Together, these components define the probabilistic forward and reverse dynamics of the diffusion process.

The forward process (diffusion) can be written as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I})$$

Here, $q(x_t|x_{t-1})$ represents the known, fixed forward (noising) process: the probability of obtaining x_t given x_{t-1} . The term $\mathcal{N}(\cdot; \mu, \Sigma)$ denotes a Gaussian (Normal) distribution with mean μ and covariance Σ . In this case, x_t is sampled from a Gaussian centered at $\sqrt{1 - \beta_t}x_{t-1}$, with variance $\beta_t \mathbf{I}$.

The reverse process (denoising) is modeled as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

Here, $p_\theta(x_{t-1}|x_t)$ denotes the *learned reverse distribution*, giving the probability of recovering x_{t-1} from x_t at timestep t . This distribution is parameterized by a neural network, which predicts the mean $\mu_\theta(x_t, t)$ and variance $\Sigma_\theta(x_t, t)$ of the Gaussian at each step. So the NN learns to predict what the mean and variance of the Gaussian distribution are using x_t and t as inputs and then randomly samples from that distribution to obtain the next slightly denoised data/image x_{t-1} . Starting from pure noise x_T , the model iteratively applies this reverse process to progressively remove noise and generate a clean sample x_0 .

DMs are mainly categorized into three categories: DDPM, Score-Based Generative Models, and Latent Diffusion Models (LDM). DDPMs, presented on [24], are the most common and serve as the foundation for many DM solutions. The main differences between these categories lie in their sampling frameworks and prediction targets. DDPMs use a stepwise probabilistic process where the model typically predicts the amount of noise added at each timestep. Score-based models, in contrast, estimate the gradient of the data distribution, essentially learning the direction in which the input should be adjusted to resemble the ground truth more closely. This is done by assigning scores that reflect how likely each point is to belong to the true data distribution, and using stochastic differential equations (SDEs) to guide the sampling process. LDMs perform diffusion in a compressed latent space, greatly reducing computational cost, and may predict either the noise or the denoised latent representation, depending on the formulation.

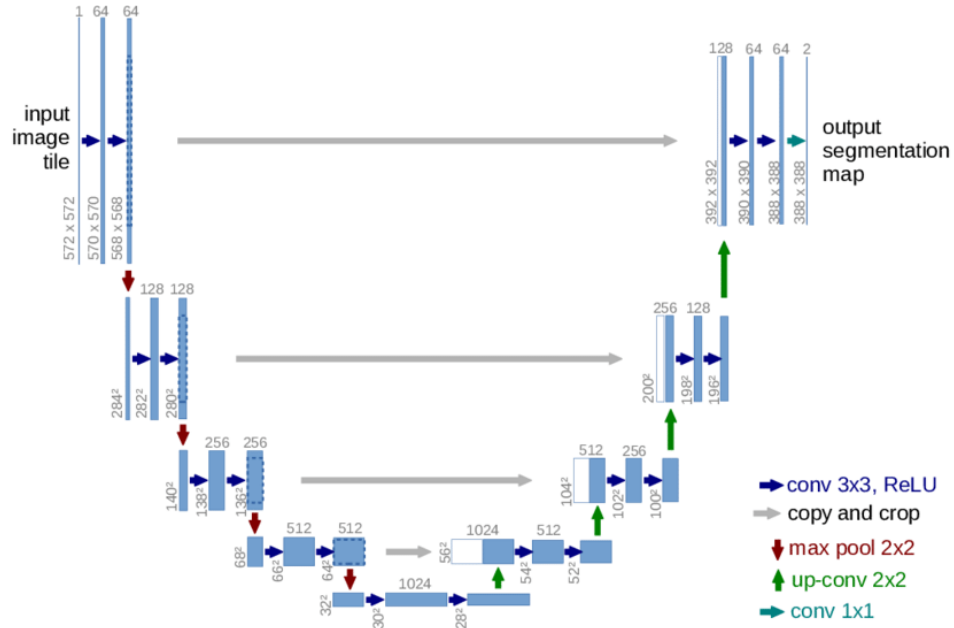


Figure 2.5: Backbone architecture of a DDPM: U-net architecture, adapted from [54]

The study that introduced DDPMs, [24], mentioned using the U-net architecture, presented in [54], as the backbone of the implementation. An illustration of the architecture is shown in 2.5, and it's possible to identify three main layers: the encoder, bottleneck, and decoder. The downsampling path (encoder) extracts multi-scale features from the input, compressing the spatial information while preserving semantic content. The bottleneck (middle) processes this compact representation and often includes time and conditioning embeddings. The upsampling path (decoder) then reconstructs the image by gradually recovering spatial details, often using skip connections from the encoder to improve reconstruction accuracy.

2.6 Evaluation Metrics

Evaluating the quality of predictions is a critical aspect of developing and benchmarking DL models, particularly in scientific applications like CFD and SR. Since models in this domain often reconstruct or generate flow fields, it is essential to define clear and relevant metrics that measure the accuracy, fidelity, and physical consistency of the output predictions. These metrics are not only used for performance evaluation but are often integrated as loss functions during training, for instance, MSE or the more specialized GMSE presented in [15].

Metrics can be broadly categorized by their focus. Pointwise metrics evaluate numerical differences between predicted and ground-truth values on a per-pixel basis. Perceptual or visual metrics attempt to assess structural and textural similarity in a way that aligns with human perception. Physics-based metrics, on the other hand, aim to evaluate whether the generated predictions respect the underlying physical laws, such as the Navier-Stokes equations.

Another important distinction is between reference-based (non-blind) metrics, which require access to ground-truth images, and no-reference (blind) metrics, which attempt to evaluate quality without any reference. Basically, blind metrics receive only the predicted images as input and assess their quality. Non-blind metrics require both the predicted and the reference/ground-truth image, and the result is a value attributed to the comparison between both images. Each metric type serves a specific purpose, and choosing the right combination depends on the goals and available data of the task, whether the emphasis is on visual fidelity, numerical accuracy, or physical interpretability.

Striking a balance between these perspectives is especially important in physics-informed DL. A visually convincing result may still violate conservation laws, while a physically accurate one may appear distorted or noisy. Therefore, combining visual and physics-aware metrics, such as using SSIM for structure and GMSE for gradient sensitivity, can offer a more holistic evaluation strategy.

2.7 Transfer Learning

Transfer learning is a widely used DL technique where a model trained on one task is repurposed for a related task, improving training efficiency and reducing data requirements. This is particularly beneficial in fields like CFD and SR, where acquiring HR data can be expensive and time-consuming. This technique is mainly applied to preserve previously obtained knowledge on tasks with a similar data distribution.

Two common strategies are feature extraction, where the pre-trained model's earlier layers are kept frozen and only the later layers are retrained, and fine-tuning, where part or all of the model is updated for the new task. The choice depends on the similarity between the source and target domains and the size of available data. A widely adopted version of feature extraction is the frozen-layer technique. Transfer learning has proven to be an effective technique in various physics-based applications [60, 77, 78]. It has also shown promise when adapted to DMs [45, 69].

By reusing generalizable features such as spatial structures or flow patterns, transfer learning allows models to generalize more effectively across domains while significantly reducing training complexity.

Chapter 3

Related Work

This chapter presents a literature review of the most relevant and recent research contributions related to the application of DL in CFD, with a focus on SR techniques and surrogate modeling. It examines existing methodologies, identifies current limitations, and highlights key advances in generative AI, physics-informed learning, and reduced-order models. Through this analysis, the chapter situates the current work within the broader research landscape and uncovers opportunities for improvement and future exploration.

Studying the literature on topics regarding SR, CFD, DMs, physics-informed models, evaluation metrics, and transfer learning reveals major gaps and possible new research avenues, and with all the relevant research articles collected, it is now possible to discuss and assess the content of the relevant scientific articles.

To this end, categorizing the studied articles into different knowledge areas would help map the current gaps in the literature. These gaps can be further studied and explored so that something innovative and/or useful can be developed, advancing the field as a whole.

The papers presented in this chapter were obtained using a systematic review strategy defined in the literature review chapter, but some of them were found with focused searches about a specific topic. All the presented papers bring new knowledge to the table and present different research paths that can be explored. It's important that the current state and strategies being used in this scientific field are fairly represented in these papers.

Our research explores state-of-the-art and reviews the literature on SR and generative modeling for CFD tasks, focusing on key datasets and commonly used data, innovative strategies, Physical conditioning strategies, evaluation metrics, and transfer learning.

For each category, the relevant problems, gaps, and current state of that knowledge area are presented with supporting papers.

3.1 Data

We can think of AI as the communication channel that interprets and translates the information within data to us humans. Fluids have always been challenging to understand and predict. The

more data available, the easier it becomes to make accurate predictions. However, obtaining accurate and comprehensive data remains a significant challenge in the field of CFD.

The variety of flows and simulations that can be performed with fluids is limitless, but the high computational cost of obtaining simulated datasets often restricts research to only a few types of situations. Fluid flow datasets span from simple and predictable laminar flows to more complex/turbulent flows and multiphase flows where interactions involving multiple interacting phases, such as liquid-liquid, liquid-gas, or solid-gas combinations, are present.

3.1.1 DAMBREAK

The DAMBREAK case is a canonical exercise to test models and study free-surface flows, often involving sudden water release scenarios like dam failures. It serves as a benchmark for testing models that handle multiphase fluid dynamics, including wave propagation, splashing, and turbulence at the interface, such as the strategy presented on [30].

3.1.2 Flow Past a Cylinder

Flow past a cylinder is a classical fluid dynamics case that focuses on the interaction of a steady flow with a cylindrical obstacle. It is widely used to study vortex shedding, wake dynamics, and pressure distribution, making it a key test case for evaluating models in laminar and turbulent regimes. Lots of work on the field, [33, 48, 49, 81], uses data on this particular CFD scenario.

3.1.3 Smoke

Smoke simulations are commonly used in fluid dynamics, [70, 76], to study the dispersion and motion of particles or gases in a medium. These datasets often involve visualizing complex flow structures, such as vortices and turbulence and are frequently applied in validating models for natural convection or mixing behaviors.

3.1.4 Kolmogorov flows

Turbulence is a widely studied topic in the field, with Kolmogorov flows often used as a benchmark example. Several studies [19, 57, 59] utilize Kolmogorov flow datasets to evaluate the effectiveness of their models in handling complex flow dynamics. They consist of a simplified model designed to study the energy cascade and the statistical behavior of turbulent flows. These flows are generated by a sinusoidal forcing term applied to the momentum equations, creating a controlled, periodic velocity field that allows researchers to examine turbulence in a more structured and predictable setting.

3.1.5 Rayleigh-Bénard convection

Rayleigh-Bénard convection is a classic example of heat-driven fluid flow, where a fluid layer is heated from below and cooled from above, leading to buoyancy-driven convection currents. This

system is frequently used to study chaotic behavior, nonlinear dynamics, and turbulence in natural and industrial processes, serving as a valuable benchmark for models addressing thermally driven flows. This scenario is used by [61, 68].

3.1.6 Aerodynamics

Airfoil optimization is a very common theme with significant potential. Datasets are critical for validating SR models under specific aerodynamic conditions.

A major challenge in airfoil optimization lies in the vast range of scenarios that can be explored, as there are infinite possibilities for airfoil shapes and numerous conditions that can be configured. With this, [6, 8, 11, 72] often need to create their own datasets to ensure the data aligns with the specific parameters, conditions, and objectives required for their experiments, such as Reynolds numbers, angles of attack (AoA), or compressibility effects. This custom approach is common in aerodynamics research, as publicly available datasets may not cover the targeted scenarios or resolutions needed for the analysis.

The Airfrans dataset, [8], is one publicly available dataset that demonstrates fluid flow interactions with airfoils. It is composed of a collection of fluid flow simulations around NACA airfoils, generated using the Navier-Stokes equations and the RANS approach in OpenFOAM, providing point-cloud data with variables such as velocity, pressure, and turbulent kinematic viscosity for a range of Reynolds numbers and angles of attack in subsonic flight regimes.

3.1.7 Bubbly flow

Bubbly flow refers to a two-phase flow system where gas bubbles are dispersed in a continuous liquid phase. It is often studied in configurations like bubble columns due to its applications in chemical, biological, and industrial processes. Bubbly flow datasets are used to evaluate specific physical phenomena like gas-liquid interactions. The [41] trains GANs proposed for natural images on a bubbly flow experimental Fluid Dynamics dataset, the configuration involves capturing high-speed images of bubbles in various liquids under controlled flow rates and gas velocities in a bubble column setup. The dataset used is private. Studying bubbly flow is useful because it provides insights into gas-liquid interactions, which are critical for optimizing processes like mass transfer, heat exchange, and mixing in diverse engineering applications.

3.1.8 SuperBench

SuperBench was introduced by [51], and it's a benchmark dataset for comparing spatial SR methods in scientific applications. The research value of this dataset comes from the larger HR dimensions it offers and the added diversity of scientific domains.

SuperBench is composed of 5 different datasets, with three different scientific themes, as shown in Figure all of them present a challenge for SR methods because of the inherent multi-scale structures of such domains. It includes two datasets for fluid flows, with different Reynolds

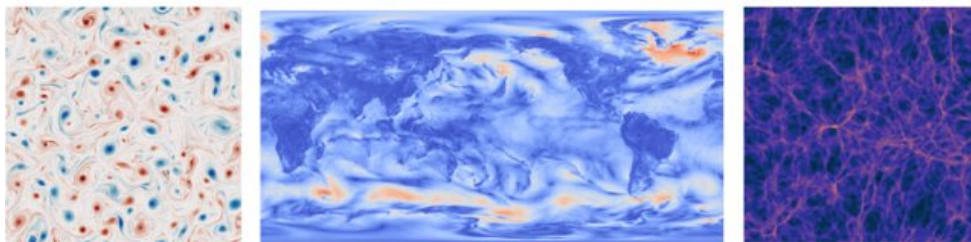


Figure 3.1: SuperBench samples from three different scientific domains, adapted from [51]

numbers, 16000 and 32000. A public dataset for weather data is also included, as well as two datasets for cosmology data, one LR dataset, and another HR dataset.

3.1.9 MegaFlow2D

The MegaFlow2D dataset [74], is a large-scale, multi-fidelity resource designed for ML SR tasks in CFD. This public dataset has data on three types of scenarios: external flows around circles and ellipses and internal flows (through nozzle-shaped geometries). The data is very parameterizable, and this will enable the development and benchmarking of ML models for SR tasks while providing more customizable configurations and aiding models to avoid overfitting specific tasks.

3.1.10 Discussion

Most datasets are simplified and synthetic, generated through simulations under controlled, canonical configurations such as 2D Kolmogorov flows, flow past a cylinder, or Rayleigh-Bénard convection. These scenarios, while useful for benchmarking, lack the complexity and variability of real-world, three-dimensional, and multi-physics flows. Furthermore, the overwhelming majority of data comes from DNS or RANS simulations, with very limited experimental data available. This reliance on synthetic data makes it difficult to assess how well SR models generalize to real-world conditions, where noise, measurement errors, and unmodeled physics are present.

A further challenge is the two-dimensional nature of many datasets. Although 2D data simplifies computation and visualization, real fluid flows are inherently three-dimensional, and SR models trained exclusively on 2D data may fail to capture essential features of 3D dynamics, such as vortex stretching or fully-developed turbulence. The number of samples in publicly available datasets is also often limited, which constrains the ability to train large models effectively. Some datasets offer only a few hundred or thousand samples, which is small compared to the data-hungry nature of modern DL models. This makes data augmentation or synthetic degradation necessary but also risks overfitting to simplistic scenarios.

Finally, the lack of diversity, in terms of flow regimes, geometries, boundary conditions, and physical parameters, limits the robustness of SR models. When trained on narrow distributions of flows, models may struggle to generalize to unseen conditions, which is critical for practical applications.

In summary, most current datasets are synthetic, simplified, low in diversity, often 2D, and limited in size. These factors impact not only how well SR models can be trained but also how well they generalize. Future work should focus on expanding datasets toward more complex, 3D, and experimentally-derived flows, with higher sample counts and richer diversity in scenarios and conditions, in order to enable more robust and physically meaningful SR models.

3.2 Innovative generative models and strategies for fluid flow predictions

The strategies to predict fluid flows are constantly undergoing changes, and new innovations are being applied and tested. The current trend of the field is very oriented toward the use of generative models and similar strategies for a multitude of tasks, with the objective of improving the efficiency of CFD simulations.

SR is a primary focus, but exploring other approaches could provide insights and opportunities to enhance state-of-the-art strategies. As mentioned, the main focus is generative models, which encompass DMs, GANs, CNN-based strategies, and other DL-based approaches.

3.2.1 CNN-based strategies

CNNs are also a very commonly used strategy for this kind of problem because these kinds of networks deal very well with grid-like data. A common theme within the field is the application of aerodynamics and the study of airfoils and airflows. [6] compares DeepCFD, a CNN-based surrogate model, with CFD simulations and particle image velocimetry (PIV) measurements for variant airfoils. The study showed the potential of the CNN-based strategy for rapid aerodynamic analysis. Along similar lines, [11] compared data-driven techniques, specifically CNNs, Geodesic CNNs (GCNNs), and Proper Orthogonal Decomposition (POD), for predicting compressible flow fields over the transonic RAE 2822 airfoil. Transonic flows refer to speed ranges where the air-flow over a body includes both subsonic and supersonic flow regions. These kinds of flows are complex to model. The CNN approaches demonstrated both efficient and accurate predictions, whereas POD struggled to achieve similar accuracy, especially in capturing complex transonic flow behavior.

Beyond aerodynamic applications, CNNs have also proven valuable in biomedical scenarios, where fluid dynamics play a critical role. [20] applies a CNN-based network called 4dFlowNet applied to Magnetic Resonance Imaging (MRI) to improve blood flow assessment while reducing reliance on expensive MRI scans. The approach tries to upsample, through SR, velocity fields of noisy and LR MRI data with promising results. The need for extensive amounts of data, HR data specifically, limits the predictive capability of DL models. [21] builds on this notion and proposes a CNN model capable of performing SR in domains relevant to cardiovascular and biomedical applications. This approach manages many HR enhancements with impressive speed-ups when compared to CFD simulations, without the need for HR training data.

To reconstruct turbulence flow field images with LR, [81] proposed SRUnet, which combines features of two CNNs: SRCNN and U-net. This model was validated with small datasets and was shown to deliver high-precision results when applied to the velocity field of the two-dimensional (2D) cylinder wake flow. To further evaluate the model, it is also applied to a hydrofoil velocity field reconstruction task, where good accuracy is also achieved, demonstrating the generalization capability of CNN-based SR models.

3.2.2 GAN-based strategies

GANs are likely the most popular choice to generate realistic HR data from LR inputs and very often present good results as surrogate models on fluid flow field predictions.

The work, [30], applies a modified SRGAN model to upsample fluid phase fraction in multi-phase turbulent flows, and the model significantly outperforms non-ML upsampling methods and can preserve a substantial amount of detail. GANs can also be applied to 3D simulations, such as volumetric floating smoke cases. [70] develops a method for 3D SR using GANs, called multi-pass GAN, by decomposing volumetric data generation into more manageable orthogonal passes with respectable large volumetric up-scaling factors. The authors also state that GAN models are suitable for SR because they perform effectively on problems with multiple solutions.

Another interesting approach using GANS when training data is sparse applied to the task of predicting the flow field structure for supercritical airfoils, is presented on [72]. The method proposed is called data augmented Generative Adversarial Network (daGAN) and basically relies on the use of two GANS, one to estimate the distribution of the available sparse training data and another with a focus on data augmentation. This type of approach that reduces the need for huge amounts of data can be very valuable in this field. Spatiotemporal tasks are hard, especially when the space is three-dimensional. [84] presents another double GAN approach for such a task applied to turbulent/swirling flames. The devised approach uses one GAN for the spatial upsampling and another for the temporal upsampling, achieving superior results when compared to interpolation techniques.

3.2.3 Diffusion-based strategies

The use of DMs is becoming a very active research topic in the field, with still very few explorations of the topic. Despite this, some evidence of the potential of such technology for predicting fluid dynamics is surfacing, and curiosity is growing, with quite interesting strategies being developed.

Initially, GANs, due to their efficiency and accuracy, were widely used and explored, but DMs started proving to be a promising alternative when this type of model surpassed GANs in several common Computer Vision (CV) tasks. DDPM is the most common definition of DMs, and this architecture is presented on [24], where it shows superior results when compared to GANs in image synthesis tasks. It also shows DMs to be robust against common GAN limitations, such as mode collapse and training instabilities. Mode Collapse of GANs regards the limitation of the

generator, which converges to produce limited or repetitive outputs instead of diverse samples. The training instabilities of GANs happen when the loss function results often diverge or oscillate due to the adversarial nature of generator-discriminator dynamics.

The study [16] establishes DMs as state-of-the-art for both conditional and unconditional image synthesis tasks, improving significantly on GANs. The attempt to combine multiple DMs in a cascaded architecture, [25], also shows superior results by generating high-quality class-conditional images, taking advantage of a progressive strategy. Latent Diffusion models were also tested for unconditional and class-conditional tasks in image generation, [53], where, again, it outperformed GANs. For text-to-image generation, [23] proposes VQ-Diffusion, which joins DMs with a vector-quantized autoencoder, eliminating unidirectional bias and error accumulation typical of GANs. This approach significantly outperformed GAN-based methods in generating high-fidelity and complex multi-object scenes, offering better trade-offs between quality and speed. Unified Frameworks of DMs were also developed, [56], for several CV tasks comprising inpainting, uncropping, colorization, and JPEG restoration. The versatility of DMs for these tasks shows robustness for generating very fine details, which is really useful for turbulence modeling because of the high-wave-number structures inherent in turbulence. Multiple studies show the proficiency of DMs for modelling turbulent flows, [31], shows an approach for reconstructing turbulent rotating flows using DMs that could surpass GANs in pixel-level and statistical accuracy metrics.

In an effort to explore the potential of DMs applied to fluid flow predictions further, several methods were built to this effect. A very common recent approach relies on the combination of DMs with other popular strategies into a single framework. A combined model has already been presented, [48], where a DM (SR3) and a Residual Network (EResNet) are integrated to reconstruct turbulent flow fields. In this approach, the DM learns the distribution of the flow, and the residual network improves the accuracy of the flow field. Through a comparison of the framework's results with those of SRCNN and ESRGAN, the model shows superior capture of turbulent details. Another approach, developed by [61], takes a different methodology by applying two ML techniques sequentially, first a CNN-based surrogate model. This model approximates the entire fluid flow field based on initial conditions. Then, a DM is applied to improve the accuracy of smaller-scale flow structures. The DM was tested with Rayleigh-Bénard convection flows, and multiple types of noise were successfully handled. This approach differs in that the DM is applied in the second stage of the framework. Its main purpose is to enhance the accuracy of smaller-scale turbulent flows.

A decent number of studies that use DMs have applied them as a post-processing tool to refine the high-wave-number structures in turbulent flows. However, the tools used to capture the big flow structures differ.

One such approach is presented in [19], which applies a Hybrid Neural Differentiable Solver to 2D Kolmogorov turbulence. This framework achieves high accuracy, stability, and generalizability while maintaining computational efficiency. Building on this, [17] shows that a Conditional Neural Field Encoder can serve effectively as the first stage in similar frameworks, helping to capture

coarse flow features before refinement. This encoder compresses spatiotemporal flow data into latent representations, which are then received by the DM that learns the underlying probabilistic distribution. This strategy shows favorable results and demonstrates ConfiLD's superiority to CFD simulations, transformers, and other DL models. This principle of using DM for refining details in prediction is not only applied to turbulence, [44] applies a similar strategy to melt pool dynamics in Laser Powder Bed Fusion (L-PBF) additive manufacturing. The DM takes the prediction of a CNN and is able to predict the keyhole depth with impressive precision. A different refinement strategy is achieved on [64], where DMs are used to mitigate the accumulation of errors over time in sequential DL predictions.

Moving to the studies that rely only on the use of DMs, this is a very unexplored area in the field, with only a few successful approaches. Along the same lines as the previous paragraph, [57] proposes a framework for turbulent flows SR, but it uses two DMs. The DM in the first stage is used to recover low-wavenumber components, and then the second DM improves the output by generating the high-wavenumber components. The strong performance of this approach highlights that DMs are powerful tools for SR. Likely, the most enlightening works that showcase the potential of DMs coupled with PINN for turbulent flow predictions are presented on [59]. The DM is able to reconstruct HR data from either LR data or very sparse HR data. The dataset considered was a 2-dimensional Kolmogorov flow, and the results were very promising. Fully surrogate models that only receive the initial conditions of the flow and are able to represent the complete flow are harder to implement than SR models because there is less data from which the model can leap. [76] proposes a DM to perform that task applied to 2D random floating smoke, and a comparison with a conditional GAN, a PINN, and a U-net reveals that the DM is able to outperform the GAN and the U-net.

3.2.4 Other strategies

The field of predicting fluid flows explores multiple types of models in search of the most effective one. In this section, studies that use different models from the ones previously mentioned are discussed. A workflow for generating and improving surrogate models for CFD simulations using DL is proposed on [49]. This framework utilizes a two-step method that integrates a Principal Components Analysis (PCA)-based Adversarial Autoencoder (PC-AAE) with Adversarial LSTM networks. PhySR is a framework that leverages the convolutional LSTMs with residual blocks to refine the temporal dimension of CFD simulations. This framework is presented on [50], and the authors did ablation studies, which confirm the effectiveness of the component to produce the desired effect.

3.2.5 Discussion

DMs have emerged as a compelling alternative to traditional generative approaches like GANs and CNNs in fluid flow prediction tasks. Their unique iterative process enables them to model data

distributions with high fidelity, making them robust against common GAN issues such as mode collapse and training instability.

Unlike GANs, which often produce repetitive or biased outputs due to adversarial dynamics, DMs generate diverse and accurate representations, especially in complex scenarios like turbulence modeling. Accurately predicting the high-wavenumber structures in turbulent scenarios is a difficult challenge for generative models, and the fact that DMs are capable of performing this task with impressive results is a statement of the necessity of further exploring DMs. The ability to capture fine details, particularly in high-wavenumber structures, is critical for advancing CFD applications since the smaller scales have the most chaotic behaviors.

DMs have demonstrated versatility and adaptability across tasks, including SR, noise reduction, and hybrid frameworks, where they work in tandem with CNNs or physics-informed neural networks. The relatively limited exploration of DMs in CFD presents an exciting opportunity to develop innovative approaches that could surpass current state-of-the-art methods. By focusing on DMs, future research can address existing gaps in accuracy, generalizability, and the further incorporation of physics, paving the way for transformative advancements in the field of CFD.

3.3 Physics-informed Strategies

Predicting fluid behavior can be framed as accurately approximating the solutions to the governing equations of flows, primarily the NS equations. In general, generative models attempt to minimize a loss function representative of the difference between predicted and ground truth images. Training on physically consistent HR data helps guide the model toward learning the underlying physical behavior of the flow. However, it usually requires large amounts of HR data, which is expensive to obtain, and extensive training times become a requirement. Guiding these models by incorporating physical information and knowledge can definitely aid in producing more accurate results with reduced training complexity/resources.

Ignoring the physics-informed strategies that simply rely on the use of physically consistent HR data, there are other ways to integrate physics. The authors of multiple works, such as [61, 76], experienced high errors near the boundaries of the flows and hypothesize that it can be mitigated by introducing boundary conditions on the loss function of the model. More concretely, [76] shows results where PINN produces the most accurate predictions, but the training of such models can be difficult.

Works that incorporate boundary conditions into their strategies have shown significant improvements in physical consistency. The PhySR framework, [50], embeds an additional term into the loss function to penalize deviations from the residuals of governing equations and also explicitly encodes boundary conditions into its structure. A different approach is employed by [21, 33], where the boundary conditions are introduced as another loss term.

Other strategies take into account the NS equations in the loss functions. For example, in [14], the developed PINN embeds the incompressible NS equations into its loss function. This approach penalizes deviations from momentum conservation and incompressibility.

Boundary conditions and residuals are probably the most common terms to be added to the loss functions, but different terms can be used. [48] adds multiple physical gradient constraints into the loss function of the residual network, but [20] uses only the velocity gradient loss term to ensure smooth and accurate predictions near vessel walls. An interesting approach is through adding a divergence-free regularizer into the loss function to ensure incompressible flow predictions, such as the one presented on [68]. Choosing what terms to include in the loss function is an important design decision. For example, [30] experimented with loss constraints based on the Volume of Fluids (VOF) method, which enforces conservation of mass and interface sharpness in multiphase flows. While this is physically meaningful and can theoretically improve predictions in regions of phase boundaries, their results showed no performance gain. This may be because the added constraint was either redundant with existing features learned by the network or imposed too rigid a penalty that hindered optimization. Such findings highlight that while physics-based terms can improve physical plausibility, they need to be carefully weighed and compatible with the data and model to avoid counterproductive effects.

To introduce physics, there's no need to do it through the loss function, [59] incorporates the NS equations through the residual gradient as a conditioning mechanism during the diffusion process. NS equations can also be incorporated in the form of fixed convolution operations, as presented on [19].

A more indirect form of physical conditioning can be performed by pre-processing the training data. This is done in [57], where spectral filters separate high from low wavenumber structures in turbulent flows. Input features can also be used, [6] feeds the model with signed distance fields and flow region channels that help the model encode geometric and boundary condition information. The approach used in [49] leverages PCA to ensure the conservation of mass. Feature-wise Linear Modulation (FiLM), presented in [47], conditions a neural network by applying learned scaling and shifting to intermediate feature maps based on external inputs. This technique can be used to condition the model with physical information,

3.3.1 Discussion

Incorporating physical information into ML models for fluid flow prediction has demonstrated clear benefits, including enhanced accuracy, improved physical consistency, and reduced dependence on extensive HR datasets, such as the solution presented in [59]. By embedding governing equations such as the NS equations and boundary conditions directly into loss functions, leveraging architectural conditioning, or applying pre-processing techniques, these models achieve more reliable and generalizable predictions. Physical conditioning offers a promising path toward more efficient and accurate CFD solutions.

However, several gaps remain in the current literature. There is a limited exploration of combined approaches that simultaneously leverage multiple physical constraints, such as gradient regularization, divergence-free conditions, and residual-based penalties. Furthermore, boundary condition enforcement remains a significant challenge, particularly in regions with steep gradients or

intricate geometries, because the physical solution is highly sensitive there, and the geometry is hard to represent accurately.

To advance the field, future research should prioritize hybrid approaches that integrate multiple physical conditioning techniques alongside computational optimizations to make these methods scalable and efficient. Exploring innovative mechanisms for boundary condition encoding, generalization across diverse datasets, and improved data efficiency will be essential for the next generation of physics-informed ML models. These efforts will further cement the role of physical conditioning in achieving robust, accurate, and cost-effective solutions for fluid flow predictions.

3.4 Metrics for evaluating generated images

Defining ‘accuracy’ is a prerequisite for developing strategies to predict fluid flows effectively. Evaluation metrics to measure the quality of generated images are the main tool, and different assessment strategies can be employed. Evaluating the model’s results through pointwise accuracy is the main strategy, but it may not fully capture the image/grid quality. It would also be helpful to evaluate the image in terms of the quality of human perception or in terms of how physically consistent it is.

3.4.1 Pixel-level metrics

Focusing on pixel-level metrics, Mean Absolute Error (MAE), MSE, Root Mean Squared Error (RMSE) or L2 Norm, Reconstruction Error, and Relative Error are without a doubt the main metrics utilized as evidenced by [6, 11, 14, 17, 19–21, 30, 33, 44, 48–50, 57, 59, 61, 64, 68, 70, 72, 76, 81, 84]. Peak Signal-to-Noise Ratio (PSNR) is also a very common metric, [81, 84], that measures the quality of reconstructed images by comparing the signal (ground truth) to the noise in predictions.

These metrics are helpful because they are easy to compute, widely used, and provide a clear numerical measure of fidelity at the pixel level. However, they also have important limitations. They may fail to capture perceptual or structural differences that are important for human interpretation or physical realism, unlike other specific metrics addressed in the following sections. For example, an image can have similar pixel values but exhibit visually or physically unrealistic patterns. Furthermore, these metrics treat all pixels equally and ignore spatial context or higher-order correlations, which can be crucial in fluid dynamics and other structured data.

3.4.2 Perceptual metrics

Perceptual metrics evaluate the quality of SR images based on human expectations, determining whether the generated images are visually consistent with what humans perceive as plausible. The work [32] proposes Free Energy Adjusted Peak Signal-to-Noise Ratio (FEA-PSNR), a computationally efficient metric that evolves PSNR to consider the perceptual complexity of images.

Another interesting metric is Learned Perceptual Image Patch Similarity (LPIPS) presented in [83]. This metric takes advantage of deep neural networks trained using the Berkeley-Adobe Perceptual Patch Similarity (BAPPS) dataset, containing over 484,000 human judgments across traditional and CNN-based distortions, and it was able to outperform traditional metrics in terms of human judgments.

Feature Similarity Index (FSIM), [82], evaluates the similarity between a distorted and reference image based on phase congruence and gradient magnitude, capturing structural and contrast features aligned with human visual perception.

All the previous metrics need access to the reference image to evaluate the perceptual quality, but the Natural Image Quality Evaluator (NIQE) proposed in [38] is a completely blind metric. NIQE is able to assess image quality by constructing a quality-aware feature space derived from pristine, distortion-free images and determining the distance between both images.

In the field of fluid flow predictions, the Structural Similarity Index (SSIM) is a popular metric, used for example, in [81, 84], which assesses the perceptual similarity between predicted and ground truth images by capturing structural consistency.

3.4.3 Flow field-specific metrics

For physical environments, there should be metrics that assess the physical realism and adherence to governing equations of images. There are still very few metrics for this purpose, but some papers try to implement their own metrics.

Equation loss is a metric used by [59] implemented with the purpose of evaluating the normalized residual of each predicted result. This is done to measure the alignment with the Navier-Stokes equations. Another method is evaluating the divergence of the final prediction, [20, 68], meaning the incompressibility in fluid flow is quantified. Both metrics offer meaningful advantages over pixel-level measures, as they ensure the outputs are not only visually plausible but also physically valid. However, they have limitations: equation loss depends on access to and correct formulation of the PDEs, and divergence is only relevant for incompressible flows. Furthermore, neither captures perceptual quality or other structural aspects beyond the specific physical criteria they test.

Some studies, [17, 19, 57, 59, 68], use metrics that focus more on assessing turbulent flows using comparisons of PDF or Turbulent Kinetic Energy (TKE) Spectra. TKE describes how energy is distributed across different scales (or wavelengths) in a turbulent flow. These kinds of metrics, being used for qualitative approaches, are not reliable enough to generalize.

A recent and innovative loss function introduced in [15] was specifically designed to improve DL models applied to CFD tasks. The study addresses the inefficiency of traditional loss functions, such as MSE, caused by treating all pixels equally and failing to capture the localized, high-gradient features crucial in CFD fields. To overcome this, GMSE was developed to emphasize regions of the image with high spatial variance by dynamically assigning greater weight to

gradient-rich areas during training. The results presented in the paper demonstrate faster convergence and better structural accuracy yield of GMSE, making it a tool with high potential to be explored in the context of this dissertation.

3.4.4 Discussion

The evaluation of generated images in fluid flow prediction plays a crucial role in determining the accuracy and reliability of ML models. While pixel-level metrics such as MAE, MSE, RMSE, and PSNR remain the most commonly used quantitative tools, they only assess pointwise accuracy and fail to capture the perceptual or physical quality of the results. On the other hand, perceptual metrics like SSIM, FSIM, and LPIPS measure visual consistency with human expectations, offering insights into structural and textural alignment. However, these metrics are designed for general image quality assessments and are not inherently tied to the physical laws governing fluid flows, making them less reliable for physics-heavy applications.

The lack of robust, standardized physical metrics is a significant limitation in this domain. Current methods, such as equation loss, divergence evaluation, and comparisons of PDFs and Turbulent Kinetic Energy (TKE) spectra, attempt to assess the physical consistency of predictions. While they provide some insights into adherence to governing equations or turbulent behavior, they are often context-specific, qualitative, or limited in generalizability.

Although perceptual metrics are primarily designed to evaluate visual quality based on human perception, they can still provide useful insights in fluid flow scenarios. This is due to a certain paradox that happens when images with good results in pixel-level metrics can show poor qualitative analysis, as shown in [41]. Perceptual metrics can help bridge this gap.

Future research should aim to expand the domain of physical metrics. In this work, discovering an implementation of a general physical metric or integrating evaluation frameworks such as [2, 51] with fluid flow field prediction models could be interesting paths. Additionally, there's a gap in evaluation metrics for turbulent scenarios, and there's a lack of exploration of the combination of perceptual and physical evaluation metrics. The idea behind this combination is to reliably evaluate generated fluid flows while considering both perpetual quality and physical fidelity, ensuring the results are realistic and physically realizable. The GMSE loss function appears to be a strong approach for the task at hand and deserves further experimentation, but, potentially due to how recent the study is, it is still not widely used.

By addressing these gaps, the research community can ensure that ML models for fluid flow predictions are evaluated not only for their visual accuracy but also for their physical fidelity, ultimately enhancing the reliability and applicability of these models in real-world scenarios.

3.5 Transfer Learning

Transfer Learning has become a highly useful and widely accepted strategy in the modern DL field. This technique is known for allowing models to achieve faster and better results by simply reusing the weights of other similar models previously trained on similar tasks. This way, we can

take advantage of the knowledge acquired by DL models during training on related problems with slightly different data domains and distributions. It is possible to speed up the model's convergence and reduce computational demands and the complexity associated with achieving reliable results in data-scarce settings.

The possible benefits of this technique are quite obvious when extrapolating to CFD tasks since the common problems associated with CFD, such as data scarcity, training complexity, and slow convergence, can be mitigated by exploiting and experimenting with transfer learning techniques, as is stated in [26, 45, 69, 77]. It's possible to extrapolate that transfer learning can be valuable for SR tasks since it is only needed to transfer the knowledge about latent feature extraction and only retrain the layers responsible for upsampling.

As mentioned in [26], transfer learning can be performed using different methods: Full model fine-tuning, where the pre-trained model weights are updated end-to-end on the target domain. Frozen-layer transfer, where only some of the original pre-trained layers are kept fixed, and the others are retrained. Finally, adapters or lightweight modules that can be trained are inserted into the frozen backbone of a pre-trained model.

To select which strategy to choose, the trade-offs associated with each strategy need to be studied. As stated in [26], Full fine-tuning can lead to the problem of catastrophic forgetting, where all the weights are overwritten due to the target task requiring different knowledge from the one used during pretraining. Frozen-layer transfer, in contrast, is simple and effective and mitigates catastrophic forgetting by preserving the earlier layers' knowledge, but it may result in an overly biased model if the pre-trained features do not suit the target task.

Transfer learning is becoming increasingly common in SR workflows, and, as concluded in [60], there's still a lack of applications of this technique where simulated data is combined with real measurements or when domain-specific datasets are too small to train from scratch.

For DMs specifically, the research is still very embryonic, especially when applied to CFD domains. One example is [45], where a novel transfer learning method was developed capable of completely avoiding parameter fine-tuning by guiding the diffusion process of the pre-trained DM and correcting the distributional differences using a domain classifier, and therefore improving robustness under limited target data availability. Another similar method, called Transfer via Adversarial Noise, is presented at [69] and enables few-shot adaptation of DM using minimal data. The main idea is similarity-guided training with a domain classifier to guide the model toward the target domain and uses adversarial noise selection to focus learning on the most challenging noise patterns. This strategy allows effective transfer with minimal fine-tuning.

Focusing more on CFD domains, the paper [78] presents a 3D ESRGAN framework that uses a transfer learning technique based on tricubic interpolation for an SR task of turbulent flows. The developed approach involves first training the model on LR and interpolated flow data and then adapting it to HR targets. By freezing parts of the generator and fine-tuning the rest, they manage to reuse learned representations effectively across different Reynolds numbers. The premise behind the study of transfer learning by [77] is the lack of generalization accuracy of ML models, which is insufficient for small-scale datasets, in this case, for effusion cooling, a CFD domain. The authors

propose a neural operator that assimilates numerical and experimental data by first learning a shared feature space from simulation data and then fine-tuning a separate decoder using sparse experimental measurements. The reported results show that transfer learning significantly reduces the requirements for the number of training samples and training epochs to achieve the same accuracy.

These approaches demonstrate how spatial and resolution differences across datasets can be bridged effectively through modular transfer learning architectures. Notably, they support the notion that low-level representations are often transferable across domains and can be reused via frozen layers.

3.5.1 Discussion

Reflecting on the current applicability of this technique, it clearly opens new avenues for further development and enhancement, particularly in the application of DM within CFD domains. There's also definitely a knowledge gap in the application of frozen-layer transfer learning techniques with DMs

The research done demonstrates a consistent move toward parameter-efficient transfer learning in both generative modeling and scientific applications. Strategies such as frozen-layer transfer, partial fine-tuning, and guided diffusion processes offer practical solutions to the challenges of limited data, heterogeneous domains, and high computational cost.

Due to the diversity of data modalities and data structures, and in the context of most problems within CFD, frozen-layer transfer learning offers simplicity, pragmatism, and theoretical grounding. Freezing layers minimizes the risk of catastrophic forgetting, which is particularly critical given the expensive and limited quantity of HR grid-based training data.

3.6 Final remarks of literature review

The scientific articles and documents mentioned in this chapter were thoroughly analyzed, and all the relevant and common information was extracted. This was done so that the identification of patterns, limitations, common difficulties, strategies, and resources was facilitated. The main task, data/dataset utilized, type of model used, evaluation metrics, hyperparameters, applications, and results are among the extracted pieces of information.

To bring together the current state-of-the-art, an overview of the literature gaps will help identify possible solutions. First, DMs have demonstrated potential in several CV tasks, and there is a pressing need to develop more sophisticated DM strategies to be applied to fluid flow simulations and SR tasks. DMs are mainly applied in the literature as a second stage in two-staged approaches to generate fluid flow predictions, with a focus on refining the smaller scales of flows. Very little exploration of single DM strategies has been studied, especially those with physical conditioning.

The physical conditioning used in the literature lacks further study, especially on the combination of more than one strategy. The frameworks that incorporate physical attributes demonstrate

higher accuracies at the expense of additional training complexity. Possible solutions can explore combinations of boundary conditions, residuals, and divergence in the loss function or as architectural aspects of the generative models.

A significant limitation lies in the evaluation metrics and strategies. The main goal of predicting fluid flows is to obtain results efficiently that conform to the governing equations of fluids. To this effect, the current literature focuses on pixel-level evaluation metrics that are helpful to understand a model's performance but fail to account for human perceptual quality and physical feasibility. An important breakthrough would be establishing a consistent and reliable evaluation metric or framework, providing researchers the ability to correctly assess the quality of generated fluid simulation images or data.

Another significant issue in the literature is the limited diversity of flow scenarios and the scarcity of HR experimental datasets. The lack of HR datasets is explained by the computational cost associated with obtaining detailed simulations, especially for turbulent flows. The limited availability of diverse HR data restricts the training of SR models, confining their applicability to specific scenarios. The lack of datasets on diverse, realistic, and complex flow scenarios further narrows the scope of these studies, limiting their relevance to real-world applications and to the study of turbulent phenomena.

A major gap in the literature is the lack of research on transfer learning applied to both DMs and CFD domains. This gap is important to identify because there's a high amount of evidence of good results when transfer learning is applied to both CFD domains and DM when isolated. The main difficulties associated with CFD, such as a lack of HR data and the complexity of training, can be mitigated by transferring and repurposing previous knowledge.

To conclude the discussion of the literature's limitations, several promising directions for advancing the field are proposed. A critical first step is establishing robust evaluation metrics that capture both the physical accuracy and perceptual quality of results. Expanding the diversity and availability of high-resolution datasets, especially for complex and turbulent flows, could significantly improve model robustness. Combining the potential of DMs with physics-informed neural networks offers a powerful path toward more accurate and physically consistent predictions. Finally, transfer learning remains a valuable strategy to reduce data and computational demands and merits further exploration.

Chapter 4

Data & Methods

This chapter presents the core ideas, data sources, and methodological framework underpinning our experimental and analytical work. First, the selected data for the conducted work and experimentation is described. Then, a deeper presentation of the research foundations and an overview of how a complete environment is developed to explore and create all the desired hypotheses. This chapter establishes the technical framework for our results and analysis, providing the necessary context for understanding the contributions and implications of our research.

4.1 Data

As mentioned before, the AirfRANS dataset [8] is a publicly available dataset designed to support research in aerodynamic shape optimization and data-driven modeling of fluid flow. This dataset was selected for this work’s research due to its high physical reliability, ability to flow under various conditions, and relevance for practical aerodynamic applications, mainly in airfoil optimization.

The dataset contains a total of 1,000 fluid flow simulations around various NACA airfoils, each generated using the RANS equations solved via the OpenFOAM framework, so the main flow structures are retained. For each simulation, the dataset provides high-fidelity numerical results across several physical quantities, including pressure, horizontal and vertical velocity components, as can be visualized in 4.1. The simulation data is stored as unstructured meshes in VTK format, representing point-cloud solutions over the 2D airfoil domain. Each simulation consists of three mesh files (internal, airfoil, and freestream) that encode the geometry and associated scalar and vector fields, enabling spatially resolved visualization and analysis of flow behavior over a diversified range of Reynolds numbers, angles of attack, and airfoils.

To make the data usable in a context where tensors are utilized to store image information, such as most DL frameworks, the initial challenge was extracting the necessary data from the simulations and converting it into more approachable formats. The AirfRANS dataset was converted into PNG image representations of the relevant physical variables to enable its use in computer

vision-based ML models. After downloading and loading the dataset, each simulation was visualized using the PyVista library [63], which allows for intuitive rendering of unstructured mesh data. Among the available target variables, such as pressure, velocity components, and viscosity, this study focuses specifically on viscosity. This choice was motivated by the observed spatial patterns in the viscosity field, which appeared more structured and technically informative than the highly chaotic patterns in pressure or velocity fields. Interestingly, prior research has shown that DM often perform better on highly turbulent data; however, our focus on viscosity was driven by the desire to evaluate model performance in more coherent and structured flow regimes.

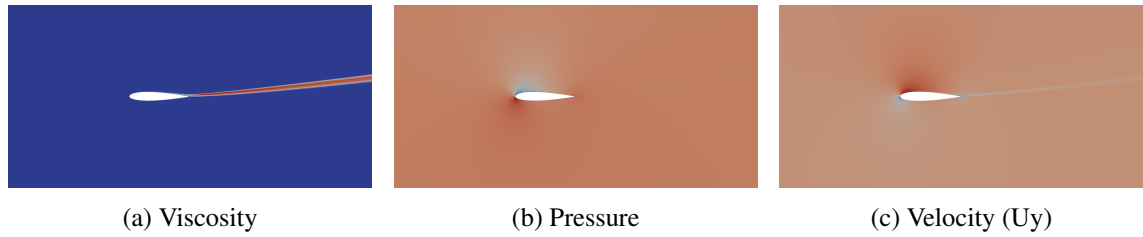


Figure 4.1: Visualization of different target variables from the AirfRANS dataset using PyVista. Each image corresponds to a scalar field plotted over the same airfoil mesh.

4.1.1 Pre-processing

From a more technical standpoint, the conversion process focused exclusively on the internal mesh file of each simulation, which contains the scalar and vector fields over the fluid domain. The preprocessing involved cropping the mesh to a consistent rectangular region to ensure spatial alignment across all samples. The target field was then visualized using either an RGB colormap for perceptual clarity or a single grayscale colormap, allowing the resulting images to be seamlessly converted into single-channel tensors suitable for DL models. Camera parameters, such as rotation, translation, and zoom, were adjusted to center the view on the airfoil and maintain a consistent perspective. Finally, the rendered image was exported in PNG format at a fixed resolution of 1500×750 pixels, yielding a standardized dataset compatible with computer vision-based learning pipelines.

4.1.2 Initial conditions

Another critical step was to save each simulation’s initial conditions and physical characteristics. The angle of attack and inlet velocity are known for a specific simulation. Still, other relevant information needed to be calculated and included, like the Reynolds number, drag, and lift forces. All the initial conditions of each simulation are saved in a JSON file.

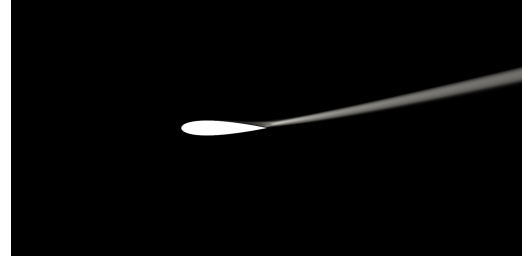
4.1.3 LR simulation data

To experiment and test the performance of DL models to upsample LR simulations, a complementary dataset formatted in ‘.cgns’ was used to obtain the necessary low-fidelity data. The dataset can

be found in [55]. The information in this dataset was computed on coarser meshes and, similarly to the original airfRANS dataset, each sample is an unstructured mesh with scalar quantities such as angle of attack, inlet velocity, and aerodynamic coefficients. The mesh files were read using PyVista, clipped to the same spatial domain as the HR data, and sampled to generate structured 2D scalar fields, interpolated linearly. An example of an illustration of the real LR data with both camps is present in 4.2



(a) Real LR airfoil simulation (RGB)



(b) Real LR airfoil simulation (Single Channel)

Figure 4.2: Comparison between RGB and single-channel real LR airfoil simulations.

4.1.4 Data extraction for transfer learning experiments

For transfer learning experiments, the objective is to evaluate the model’s ability to adapt to variations in the input data distribution. Another dataset was generated from the complementary real LR dataset that differs structurally from the original HR data to simulate this. Instead of rendering the data directly into images, the information was cubically interpolated onto structured 2D grids and stored as raw numerical arrays in .npy format. This format preserves the exact scalar values at each grid point. It allows the model to process the airfoil position and input fields with minimal pre-rendering bias, enabling a more rigorous test of generalization to unseen data modalities. A geometric mask of the airfoil was also generated using an implicit surface representation, ensuring the airfoil shape remains preserved in the output, where the points corresponding to the airfoil’s surface hold a value equal to the maximum scalar value.

4.1.5 Summary and directory structure

To conclude, here is a summary of the multiple versions of the dataset and how they are organized:

- **image_af:** HR simulations from AirfRANS converted into .png images using the `coolwarm` RGB colormap.
- **lr_image_af:** Real LR simulations derived from CGNS-format meshes, processed similarly and stored as RGB .png images.
- **grid_af:** Grayscale images of structured grids interpolated from the AirfRANS simulations, stored as .png.

- **lr_gray_image_af:** Grayscale versions of the real LR data, stored in `.png` format.
- **lr_mesh_af:** Numerical 2D grid data from the LR simulations, stored in `.npy` format for use in experiments requiring direct mesh input rather than image-based formats.

All datasets follow a consistent naming scheme and are split into training, validation, and test sets using a 70/10/20 ratio, with matching simulation identifiers across formats to ensure interoperability between modalities and experimental setups.

4.2 Methodology

The natural next step is preparing the DL environment for experimentation with DM on the previously described data sources. The initial step was to explore and study the conventions of such research coding frameworks. Finding a DM that could be an invaluable starting point for our planned research was also important.

This section describes the foundations and the possible questions our research could answer through experimentation.

4.2.1 Research foundation

As the technological foundation for this work, the [10] study and codebase were selected as a starting point. A favourable point for the decision was the exploration of multiple innovative strategies and studying the main research topics targeted by this document. From our research, this paper pioneers the effort and exploration of DM with physical conditioning for CFD-related data, and it was also the only one with an open-source code repository. With the consensual relevancy of [10] within the field’s experts, evidenced by the amount and frequency of citations by most recent studies on DM applied to CFD, it became pretty clear the need to deeply study the article and the high likelihood of it being the obvious first choice as a starting point for delving into further research.

4.2.1.1 Main task

The core task addressed in [10] is the reconstruction of HR CFD data from degraded inputs, which may be in the form of LR grids or sparsely sampled point clouds. This problem is critical in CFD research and engineering applications, as high-fidelity simulations are computationally expensive, and measurements are often incomplete or under-resolved, such as in PIV measurements. The authors reformulate this reconstruction task as a conditional generative problem, using a physics-informed DDPM to generate HR data conditioned on a noisy version of the LR input. This strategy avoids the need for paired training data and allows the model to generalize across different input distributions without retraining. The authors also found value in incorporating physical constraints on the model’s diffusion process.

The data used in their study consists of two-dimensional Kolmogorov flow simulations represented in the form of vorticity fields over space and time. Although the underlying data is spatiotemporal, the model does not account for temporal data, and therefore, the data was upsampled frame by frame, meaning that the model was designed to deal with a single image at a time; joining all the predicted frames will result in an HR video.

Several input degradation scenarios were considered to evaluate the model's robustness and generalization capabilities. These include spatial downsampling by fixed factors, $64 \times 64 \rightarrow 256 \times 256$ (4× upsampling) and $32 \times 32 \rightarrow 256 \times 256$ (8× upsampling) as well as reconstruction from sparse measurements, where only 5% or 1.5625% of the grid points are retained and the rest are missing. These diverse setups mimic practical conditions such as sensor limitations or computational constraints and test the model's ability to reconstruct fluid structure under varying levels of information loss.

4.2.1.2 Model and workflow

The introduced model builds on the DDPM framework and employs a U-Net backbone adapted for fluid flow reconstruction tasks. The goal is to learn the inverse process of a forward diffusion that gradually adds Gaussian noise to clean HR CFD data.

The forward diffusion process uses a noise schedule to control how much noise is added at each timestep. A higher beta means more noise is injected. For each training sample, a random timestep is selected, and its embedding is passed along with the LR input. The image is then noised according to this timestep; higher values correspond to stronger noise levels. This teaches the model to denoise inputs across various degradation levels, helping it learn a robust reconstruction process.

Within the U-Net, the downsampling path encodes the input into multiscale hierarchical features, while the bottleneck or middle layer leverages residual blocks to support stable and effective learning by preserving the flow of gradients, which represent how much each parameter contributes to the overall loss during backpropagation. These blocks use skip connections to help gradients propagate more effectively through the network, enabling deeper architectures and addressing the vanishing gradient problem, where gradients become too small for earlier layers to learn efficiently. Attention mechanisms are also used to enhance the model's ability to capture long-range dependencies crucial for reconstructing global structures in fluid flow. These modules are critical for resolving fine flow features and possible turbulent structures.

The upsampling path reconstructs the image, progressively recovering spatial resolution and integrating skip connections from the encoder to retain fine-grained details. Additional normalization and convolution layers at the output ensure the correct format of the output tensor. Together, these components enable the model to denoise and super-resolve CFD fields physically and visually coherently.

A key innovation is guided sampling at inference, where the model does not start from pure noise but from a corrupted version of a specific low-fidelity input. This allows the generative process to be conditioned on actual observations, making the reconstruction data-driven and context-aware. Moreover, the authors introduce an iterative refinement strategy, particularly effective in sparse input scenarios. Here, the model performs multiple rounds of sampling, each time using the previous output as a new guide. This recursive mechanism helps correct accumulated errors and improve physical coherence across iterations.

The proposed model's workflow begins with preparing HR vorticity fields obtained from DNS of two-dimensional incompressible Navier-Stokes equations. These HR fields are initially downsampled to lower resolutions (e.g., 64×64 or 32×32) or randomly sampled to simulate sparse measurements (e.g., 5% or 1.5625% of grid points), creating degraded LR inputs. Gaussian noise is added following a forward diffusion process to align these inputs with the training distribution, resulting in a noisy intermediate input x_t . The DDPM then processes this input. The model extracts features through successive convolutional layers in the downsampling path, compresses them at the bottleneck where attention blocks capture global structure and long-range dependencies, and reconstructs the data through upsampling layers with skip connections to preserve fine-grained details. In 4.3, an illustrative diagram of the model's workflow is presented.

During the network's training, the DDPM predicts a denoised HR flow field $\hat{x}_0$, which is then compared to the ground truth using a loss function that measures the difference between predicted and ground truth.

The model's Loss function uses only MSE to evaluate the model's performance during training by quantifying the pixel-level difference between the predicted and ground truth image. The study uses three complementary metrics to evaluate the quality of reconstructed flow fields. The L2 reconstruction error (MSE) measures pointwise accuracy but may overlook structural deficiencies. The PDE residual loss assesses physical validity by quantifying how well the reconstruction satisfies the Navier-Stokes equations. Lastly, the kinetic energy spectrum qualitatively evaluates how accurately the model captures energy distribution across spatial scales, reflecting the fidelity of turbulence structure. These metrics provide a comprehensive view of numerical and physical reconstruction quality.

4.2.1.3 Physics Incorporation

A significant contribution of the study is incorporating physics-based constraints in the model using different strategies. As a guidance signal, the model becomes physically aware by introducing the residual of the governing PDE, the vorticity form of the Navier-Stokes equations. The residual is computed at each time step of the reverse diffusion process and used to guide the model in two ways.

First, the vorticity residual gradient is calculated and sent as input, concatenated with the LR sample in a learned guidance setup and processed through an additional module, enabling the network to balance data-driven and physics-based correction dynamically. Basically, for each sample in the training, there's a small probability that the sample will be accompanied by the value

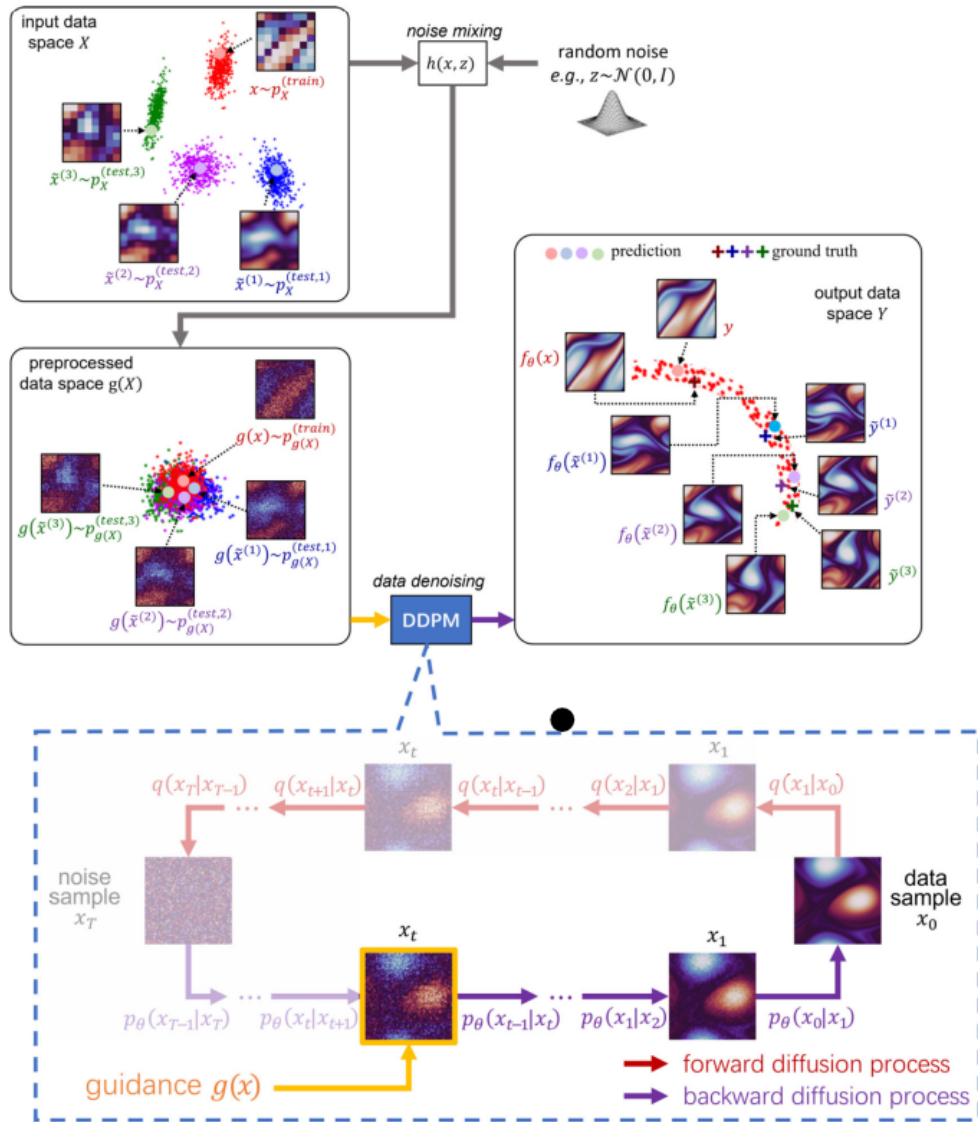


Figure 4.3: Barati workflow illustration, adapted from [10]

of the vorticity residual, which is then processed by a module that transforms the residual gradient (tensor) into a learned feature embedding, which is then directly concatenated to the input image.

The second method is applied during inference/sampling as a post-training technique, which does not change the pretrained model. Here, at each denoising step in the reverse diffusion process, the model computes the residual of the Navier–Stokes equations and takes its gradient with respect to the predicted field. This gradient is then used to adjust the prediction via gradient descent, nudging it toward a solution that better satisfies the governing physics, ensuring that the output adheres more closely to the underlying physical laws.

These methods ensure the reconstructions minimize pointwise errors and retain physically meaningful characteristics such as vorticity conservation and realistic energy spectra. Empirical results confirm that incorporating PDE residuals leads to significantly lower equation losses and more faithful recovery of multiscale turbulence features, especially in challenging setups like extreme sparsity.

For the developed experimentation benchmark, these methods rely on the use of the vorticity residual, so they were discarded. Other physical conditioning techniques were developed and tested; more details can be found in [4.2.3.2](#).

4.2.2 Core Idea

The core motivation of this dissertation is to explore the application of DMs in the context of fluid flow reconstruction. The initial objective was to evaluate how well DMs could generalize to structured physical domains such as CFD data, using the airfRANS dataset specifically, and whether they could serve as viable surrogate models for generating HR flow fields from degraded inputs. The hypothesis that DMs could offer significant benefits in this context emerged during our research, based on their impressive performance in other image generation tasks. In some cases, DMs have been shown to outperform their primary competitor, GANs, which have already demonstrated strong results in CFD simulation and reconstruction applications.

Beyond testing the raw capabilities of DMs, our research also sheds light on other interesting topics worth further investigation and several gaps in the literature, starting with integrating physical constraints into the learning process of DMs. Physical fidelity is critical in scientific ML, and this work could aid the field by further testing different physics embedding techniques, which could improve the realism and reliability of generated flow fields.

Another key area worth exploring was the effect and reliability of alternative evaluation metrics. Traditional pixel-wise losses like MSE often fail to capture perceptual or structural quality and physical fidelity. Therefore, metrics such as SSIM and GMSE were introduced to evaluate whether they offer better insights into the visual and physical quality of reconstructed flows.

The research also incorporated transfer learning scenarios to test the model’s robustness and how well knowledge transfers in DMs. By training on HR AirfRANS data and testing on real LR simulations generated from coarser meshes, the model’s ability to generalize across domains was assessed. Our research has highlighted that a frozen-layer method seems to be a practical and solid

choice in this type of problem. This setup simulated practical situations where inference must be performed on data distributions that differ from those seen during training.

Overall, the experimental plan was guided by combining state-of-the-art generative modeling with physics-aware reasoning, robust evaluation metrics, and transfer learning techniques to push the boundary of what is currently possible in data-driven CFD reconstruction and validate the strength of those many techniques. The following sections detail the specific adaptations, workflows, and configurations to implement these ideas in practice.

4.2.3 Architectural and Functional Restructuring

The following architectural and functional changes were implemented to support the experiments in this research. These modifications enhanced the baseline codebase to handle the AirfRANS dataset, incorporate diverse physical conditioning strategies, and facilitate the training, testing, and evaluation of the proposed DM models.

In the next sections, we detail these changes, as well as the experimental workflow and setup. All the development was performed using the PyTorch deep learning library, presented in [46], which was used not only for model implementation and training but also for efficient tensor manipulation and data flow management throughout the pipeline. Additionally, TensorBoard, presented in [1], was used for keeping track of training and validation losses over time. Git was used as the version control system to manage the codebase and ensure reproducibility of results.

4.2.3.1 AirfRANS data

The original codebase, developed for spatiotemporal Kolmogorov flow data, required substantial architectural and functional adaptations to support the AirfRANS dataset and the broader experimental objectives of this work. This required the addition of a specific class to handle the dataset's needed operations, such as loading, preprocessing, and item fetching.

The preprocessing steps include resizing the images, according to whether the image is LR or HR, to a quadrangular format, converting the PNG or NPY samples to a tensor, and normalizing the pixel values. Normalization has proven beneficial in speeding up DL model training at a very low accuracy cost.

The most demanding adaptation involved handling real low-resolution simulation data, which required coordinated access to multiple data sources and careful mapping between inputs and their corresponding HR targets.

Every sample retrieval provides paired HR and LR tensors and simulation-specific initial conditions such as Reynolds number and angle of attack. In more advanced configurations, the dataset includes additional physical fields such as velocity, pressure, and viscosity, either in global mean values or as complete spatial distributions. These fields are used as conditional inputs to guide the model during training or inference.

4.2.3.2 Physical conditionings

Furthermore, the model was extended to support multiple forms of physical conditioning, enabling it to incorporate simulation-specific context directly into the generative process. Three main conditioning strategies were implemented:

Timestep Embedding takes physical parameters such as Reynolds number and angle of attack into the model and through a small NN where a learned embedding is produced and added to the timestep embedding vector. With this, the model has access to new simulation-specific information associated with the timestep embedding used to determine how denoising is done.

FiLM Conditioning applies the physical parameters to modulate the internal activations of the model layers. In other words, the model uses the simulation-specific information, such as Reynolds number and angle of attack, to scale and shift the extracted/learned features within the convoluted input tensor.

The Film module result is calculated using the following equation:

$$FiLM(F) = \gamma(\mathbf{c}) \cdot F + \beta(\mathbf{c}) \quad (4.1)$$

In this equation, FiLM stands for Feature-wise Linear Modulation, a technique that applies learned, conditioning-dependent transformations to the input feature tensor F . The functions $\gamma(\mathbf{c})$ and $\beta(\mathbf{c})$ represent scaling and shifting factors, respectively, which are both functions of a conditioning vector $\mathbf{c}$. This vector can encode external parameters such as the Reynolds number, angle of attack, or other contextual variables relevant to the simulation. The modulation is applied channel-wise, allowing the network to adapt its behavior dynamically based on the conditioning input.

Explicit Channel Conditioning replicates physical parameters, Reynolds number, and angle of attack, spatially and concatenates them as additional channels to the input image. This approach provides the model with localized access to the conditioning information associated with each simulation at every pixel, allowing it to leverage these inputs in a spatially aware manner throughout the network.

Beyond these core strategies, two extended and more complex conditioning variants were developed. These strategies can be defined as evolutions of the Explicit Channel Conditioning technique.

The *exp_imp* configuration includes the global mean values of key physical fields such as velocity, pressure, and viscosity, concatenated alongside Reynolds number and angle of attack. To prevent data leakage, the mean value of the target variable is not added. This variant provides coarse but informative physical context, suitable for situations where complete spatial fields are unavailable or impractical to compute.

The more advanced *exp_imp_full* variant includes the complete spatial distributions of these fields, effectively enriching the model's input with detailed flow structure information. In both cases, careful attention was given to avoid data leakage by excluding the target variable from the conditioning channels.

4.2.3.3 Transfer Learning

Lastly, a transfer learning framework was implemented to adapt a pretrained model, trained initially on real LR image data inputs, to a cubically interpolated 2D grid. This was accomplished by freezing the main body of the network and fine-tuning only the final reconstruction layers.

Considering that the core architecture of the DM is based on a U-Net, all layers are frozen except those responsible for the reconstruction of the output image from the latent feature maps. This includes the 'u' layers, which perform the progressive upsampling and refinement of features. In addition, the final layers, which include norm_out for normalization of the final produced feature map, conv_out for final feature-to-pixel projection, and final_upsample for spatial resolution adjustment, are also kept unfrozen, allowing them to adapt during fine-tuning.

In the developed framework, the plan is to transfer the weight of a model pre-trained on a reconstruction task of HR simulation data, from LR simulation data with advanced physical conditioning. From this pre-trained model, the feature extraction knowledge is reused in the training process of a similar task, but the input data distribution is slightly different.

By reusing the feature extraction capabilities learned from real LR simulation image data and adapting only the final layers, the postulated results should demonstrate a reduction in trainable parameters and, therefore, a reduced training time. Theoretically, this reduction in trainable parameters should also minimize overfitting and facilitate domain adaptation with limited access to labeled HR samples.

4.2.3.4 Improved Workflow

As mentioned, the model follows a denoising diffusion probabilistic framework built around a U-Net architecture. An HR ground truth and its corresponding LR version are retrieved at each training step, along with any conditioning variables. A timestep t is randomly sampled, and noise is added to the LR image based on a predefined diffusion schedule, simulating a degradation process.

A significant architectural change was the addition of true upsampling capabilities. Unlike the original model, which preserved input resolution, this version incorporates learnable upsampling layers at the output stage to reconstruct HR flow fields from LR inputs.

At initialization, the model receives an object outlining all the details about the configurations of the specific run. In order to allow the model to dynamically adapt its architecture to a certain physical conditioning technique, the model class was adapted to parse the configuration file and setup the necessary architectural changes, such as correcting the number of input and output channels of some components, adding additional modules, expanding the input tensor with more channels, etc.

4.2.3.5 Loss Functions

During training, any DL model needs a value to guide and provide feedback to the model about its current performance on predictions. The loss function is responsible for informing the model

about the accuracy of its predictions, and designing an optimal one is a tough challenge.

In [10], the loss function only evaluated pixel-level differences, clearly providing room for further improvements. GMSE is introduced as a promising metric with the potential to enhance the physical and structural reliability of results produced by DL models in CFD contexts, making it a natural choice for inclusion in certain experiments. Another interesting metric to experiment with is SSIM to evaluate the structural fidelity of the predictions. The clear structures present in the images of viscosity scalars of airfRANS simulations indicate a possibility of improvement when these kinds of metrics are incorporated in the loss function.

The main types of loss functions implemented are the following:

- **MSE Loss:** A standard mean squared error between the predicted and ground truth noise, ensuring pixel-wise accuracy.
- **Composite Loss (MSE + SSIM):** A weighted combination of MSE and SSIM, balancing numerical accuracy and perceptual quality.
- **GMSE:** Evaluates the predicted and ground truth, penalizing differences in spatial gradients and emphasizing preservation of flow structures, such as boundary layers or sharp transitions.

4.2.3.6 Evaluation Metrics

To comprehensively assess model performance, during the test phase, multiple evaluation metrics were used:

- **MSE:** Measures mean squared error over the entire image.
- **SSIM:** Evaluates perceptual similarity, particularly useful for visual fidelity.
- **GMSE:** Measures alignment of spatial gradients, essential for preserving physical realism in flow structures.

These metrics are quantitative, meaning the final measurement is a numerical value. However, the qualitative analysis of the predicted images during the test or along the training phase can also provide invaluable feedback about the performance of the model, although it is a subjective judgment.

4.2.3.7 Training Setup

All models were trained for up to 300 epochs using the ADAM optimizer with an initial learning rate of 0.00001. Training was conducted both locally and on the Newton virtual machine at LI-ACC, depending on the computational requirements of each experiment. The Newton machine is

equipped with dual Intel Xeon 6258R processors at 2.7GHz, 128GB of RAM, and a Quadro RTX 8000 GPU with 48GB of memory, providing the necessary resources for high-resolution DL tasks.

The training followed a diffusion-based framework with a total of 1000 diffusion steps, using a linear beta noise schedule starting at 0.0001 and ending at 0.02. The network was configured to scale across four resolution levels according to a multiplier pattern of [1, 2, 2, 2]. Each resolution stage included two residual blocks, and attention mechanisms were applied at the 16×16 resolution level.

The architecture also included a dropout rate of 0.1 and used convolutional layers for all re-sampling operations. To stabilize training and improve generalization, an Exponential Moving Average (EMA) of the model weights was maintained using a decay rate of 0.9999.

4.2.4 Configuration

The codebase supports experiment configuration via YAML files. These configuration files define the model architecture, dataset paths, loss functions, training hyperparameters, upsampling scales, and conditioning methods. Separate configurations are used for training and inference, enabling flexible experimentation.

4.2.5 Runner

The runner module orchestrates the training and testing pipeline. It handles dataset loading, model instantiation, optimizer setup, scheduling, and the execution of training and testing routines. The runner also performs periodic logging and saving of intermediate results for visualization and debugging.

In the initial framework, the testing phase only evaluated MSE. In this project, other evaluation metrics are used, and therefore, the testing process was changed to calculate, for each testing sample, the value of all the used metrics (MSE, SSIM, GMSE) and their averaged value when compared to the ground truth.

During the testing phase, the runner was enhanced to generate and save evolution plots of the training and validation losses by extracting data from the TensorBoard logs recorded during training.

The ability to compare low-resolution input images directly with the ground truth high-resolution data, quantifying the baseline error before reconstruction, was also included. This allows researchers to evaluate not only the absolute quality of the predictions but also the improvement over the initial degraded or low-fidelity inputs when compared with the HR simulation.

4.2.6 Training

During training, the runner loads and preprocesses each batch using the AirFransDataset class, which prepares both LR inputs and their corresponding HR targets. A random timestep is selected for each sample, and Gaussian noise is added according to the beta schedule, simulating the forward diffusion process. Conditional inputs, such as Reynolds number, angle of attack, or full

```

Saved visualization: ./experiments/airfrans_run_8x_real_lr_cond_exp_imp_mse_v4/logs/./weights/airfrans_run_8x_real_lr_cond_exp_imp_mse_v4//
batch_144_7_comparison.png
INFO - diffusion_tub.py - 2025-06-25 19:47:21.248 - step: 48518, loss: 0.03732907399535179, data time: 0.011990877298208384
=====
Epoch: 144/300, Loss: 0.033917763651433314
[Validation] Avg Loss: 0.0677, MSE: 0.0340, SSIM: 0.0317, GMSE: 0.0020
Saved visualization: ./experiments/airfrans_run_8x_real_lr_cond_exp_imp_mse_v4/logs/./weights/airfrans_run_8x_real_lr_cond_exp_imp_mse_v4//
batch_145_0_comparison.png
INFO - diffusion_tub.py - 2025-06-25 20:08:24.051 - step: 48529, loss: 0.03125523030757904, data time: 0.3783609867095947

```

Figure 4.4: Example of logging during a training stage

physical fields, are formatted based on the experiment setup and passed to the model along with the noisy input and timestep.

The model performs a forward pass to predict the clean signal, and the result is compared to the ground truth using the configured loss function. Gradients are computed and used to update the model parameters with the Adam optimizer. Model checkpoints are saved periodically to preserve training progress.

4.2.7 Validation

Although validation was not originally implemented in the codebase, a mechanism was added to track validation loss at the end of each training epoch. After each epoch, the current model is evaluated on the validation set, which represents 10% of the full dataset, and the average value of all evaluation metrics is computed and recorded.

4.2.8 Testing

During testing, the trained model is loaded and applied to the test set, where each LR input is noised using the final timestep of the diffusion schedule to simulate maximum degradation. The model then performs the reverse denoising process, conditioned on relevant simulation parameters, to reconstruct the HR output. For each sample, MSE, SSIM, and GMSE metrics are computed against the ground truth, and their averages are reported. Additionally, visual comparisons of inputs, predictions, and targets are saved, and if enabled, training and validation loss evolution plots are generated from the TensorBoard logs to support post-training analysis.

4.2.9 Logging

Logging is a key component of any experimental pipeline; an example of this process within the scope of this project is presented in Figure 4.4. Throughout training, the runner, depending on the logging frequency parameter, logs training metrics such as loss values and data loading times using TensorBoard, enabling live tracking of model performance.

At regular intervals, visual comparisons between predicted outputs and ground truth images are generated and saved to disk, roughly eight images per epoch. These visualizations allow qualitative assessment of the model’s reconstruction capabilities as training progresses.

This comprehensive logging setup facilitates in-depth performance comparison across different experiments, configurations, and model variations.

Chapter 5

Experiments

This chapter presents and discusses the research experiments carried out throughout the project. These experiments were designed with varying objectives, but all share the overarching goal of generating meaningful insights into the performance and limitations of DMs in SR tasks for fluid simulation data. The computational tools, training infrastructure, and implementation details supporting these experiments are described in Chapter 4.

5.1 Overview of Experimental Goals

The experiments were designed to address key research questions by testing different conditioning methods, loss functions, and input types. The influence of the physical parameters was evaluated, metrics like MSE, SSIM, and GMSE were compared, and the performance of DM was analyzed on both synthetically downsampled data and real LR data. Transfer learning was also explored to assess model generalization.

A central aim of the experimental benchmark was to systematically test a wide range of DL strategies. These strategies were selected to explore both potentially promising techniques and methods that, if ineffective, could be ruled out for future work. In this way, both successful and unsuccessful results would contribute to a clearer understanding of the design space.

As an initial exploration and familiarization with DMs, we began by reproducing the results from [59], using their original codebase. This baseline experiment involved training a physics-informed DM on Kolmogorov data using the authors’ default settings and pre-configured environment. The goal was to validate our setup, study the code, and identify the key architectural and implementation aspects requiring adaptation. These modifications are described in detail in Chapter 4.

Following this initial validation step, the experimental pipeline was adapted to work with the airfRANS dataset. This marked the beginning of the main research contributions.

5.2 Baseline experiments

The first set of experiments aimed to evaluate the baseline performance of DMs applied to the task of image SR within the context of fluid dynamics. These experiments were conducted using the airfRANS dataset, which consists of HR CFD simulation outputs. The primary goal was to establish a set of reference performance benchmarks under no physical restraints, while also assessing the effect of different loss functions on the quality of the generated outputs.

The simplified training workflow for these experiments, focusing on a single data sample, begins by selecting an HR image representing the viscosity field from a specific simulation. This image is then downsampled by a predefined factor to generate the corresponding LR input. The LR image is fed into the DM, which attempts to reconstruct the original HR image. The model's prediction is then compared to the ground truth HR image using a chosen loss function, allowing evaluation of the reconstruction accuracy and guiding the optimization process during training.

Models were trained to perform upsampling at two different scaling degrees, 4× and 8×. To generate the corresponding LR inputs, HR images were synthetically downsampled using a bilinear interpolation resizing method. This technique was chosen to simulate resolution degradation in a controlled, consistent, and run-time manner.

For each upsampling factor, two separate training configurations were explored. One model was trained using MSE as the sole loss function. MSE provides a pixel-wise measure of difference between the predicted and ground truth images, emphasizing intensity accuracy but often failing to capture perceptual similarity. To verify the impact of incorporating metrics that capture structural similarity, a second model was trained using a combined loss function that incorporated both MSE and the value of (1-SSIM) with equal weight.

Quantitative results for both loss function configurations are presented in Tables 5.2 and 5.1. Table 5.2 summarizes the results for models trained with the combined MSE and SSIM loss, while Table 5.1 shows the outcomes for models trained exclusively with MSE. In both tables, lower values of MSE and 1-SSIM indicate better performance, as they correspond to reduced numerical error and increased structural similarity, respectively.

For a qualitative analysis, an example of the result of the 8x upsampling model is provided in 5.1. As can be observed, the models' upsampling results are quite impressive, with a few imperfections.

Table 5.1: Comparison of different upsampling levels using MSE loss only

Upsampling Level	MSE ↓	1-SSIM ↓	GMSE ↓
4×	3×10^{-4}	7.9×10^{-3}	4×10^{-4}
8×	5×10^{-4}	0.0150	6×10^{-4}

After training the baseline models, the loss evolution was recorded to illustrate the learning dynamics. Figure 5.2 shows the curves for the 4x and 8x upsampling experiments with MSE and MSE+SSIM losses. In all cases, the loss drops quickly at first and then converges more slowly,

Table 5.2: Comparison of different upsampling levels using MSE + SSIM loss

Upsampling Level	MSE ↓	1-SSIM ↓	GMSE ↓
4×	4×10^{-4}	0.0118	4×10^{-4}
8×	6×10^{-4}	0.0177	6×10^{-4}

indicating stable training. The MSE+SSIM loss stabilizes slightly higher, reflecting its balance between perceptual and pixel-wise accuracy.

5.3 Physical conditioning

Three different conditioning strategies were investigated to evaluate the effectiveness of physical conditioning in DMs applied to the airfRANS dataset. The first approach employed a timestep embedding, which incorporates the initial conditions into the timestep embedding sent along with the LR image into the model’s input representation. The second strategy used FiLM conditioning, which modulates intermediate features in the network based on external physical parameters. The third approach was an explicit conditioning method, where an additional input channel was added to each sample, containing constant values representing the initial physical conditions of the simulation.

All three conditioning methods were evaluated using two distinct loss function configurations: one using only MSE, and another linearly combining MSE with the SSIM. The physical parameters used for conditioning, mainly the Reynolds number of the fluid and the airfoil’s angle of attack, serve as contextual inputs to guide the model during training. The objective is to allow the model to dynamically adapt to different flow regimes and simulation scenarios.

The results are summarized in Tables 5.4 and 5.3. As observed, the timestep embedding strategy consistently outperformed both FiLM and explicit conditioning, yielding the lowest error rates across both MSE and 1-SSIM metrics.

Table 5.3: Comparison of physical conditioning methods using MSE loss

Conditioning Method	MSE ↓	1-SSIM ↓	GMSE ↓
Timestep Embedding	2.5×10^{-3}	0.0328	1.1×10^{-3}
Explicit Conditioning	5.5×10^{-3}	0.0358	1.2×10^{-3}
FiLM Conditioning	0.0168	0.0355	1.4×10^{-3}

Table 5.4: Comparison of physical conditioning methods using MSE + SSIM loss

Conditioning Method	MSE ↓	1-SSIM ↓	GMSE ↓
Timestep Embedding	2.4×10^{-3}	0.0304	1.1×10^{-3}
Explicit Conditioning	7.9×10^{-3}	0.0386	1.3×10^{-3}
FiLM Conditioning	0.0178	0.0370	1.4×10^{-3}

Among the three strategies, timestep embedding demonstrated the most promising performance, both quantitatively and qualitatively. However, despite expectations that conditioning

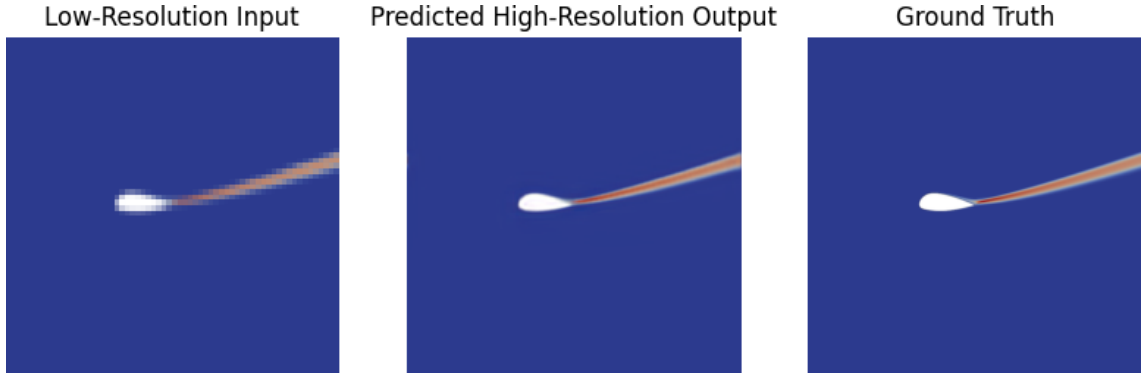


Figure 5.1: 8 $\times$ Upsampling without Physical Conditioning.

would lead to better results than unconditioned models, none of the methods tested surpassed the baseline performance without conditioning.

In particular, the explicit and FiLM-based conditioning approaches appeared prone to overfitting low-frequency background features, such as uniform color regions, rather than capturing the meaningful physical variations present in the data. This likely led the models to settle into local minima, hindering their ability to generalize across flow scenarios. These findings indicate that, while the inclusion of physical parameters is theoretically beneficial, the method of conditioning must be carefully designed to avoid introducing misleading or redundant signals.

Furthermore, while SSIM is generally regarded as a perceptual quality enhancer, its contribution was mixed in this context. Notably, incorporating SSIM into the loss function slightly improved results for timestep embedding, but had minimal or even negative effects on the other two methods. This suggests that perceptual metrics like SSIM may only be beneficial when the model architecture is able to align them with physically meaningful representations.

For visual analysis, Figure 5.3 shows the output of a model using a timestep embedding with the MSE + SSIM loss configuration. Visually, the timestep-conditioned result appears sharper and more coherent, supporting the quantitative findings.

An analysis of the training dynamics was done using the loss evolution curves during training, presented in Figure 5.4. For Timestep Embedding, the loss decreases quickly and stabilizes at a low value, showing stable and effective learning. Adding SSIM slightly slows convergence, but the curve remains smooth and stable. With Explicit Conditioning, the loss decreases abruptly at the beginning as the model quickly captures basic patterns. However, it settles at a higher value than Timestep Embedding. The MSE+SSIM loss is noisier and remains slightly higher than MSE alone. For FiLM Conditioning, the loss decreases the slowest among the three strategies and settles at the highest values, especially with MSE+SSIM. This indicates that the FiLM modulation may have struggled to align with the task in this setup.

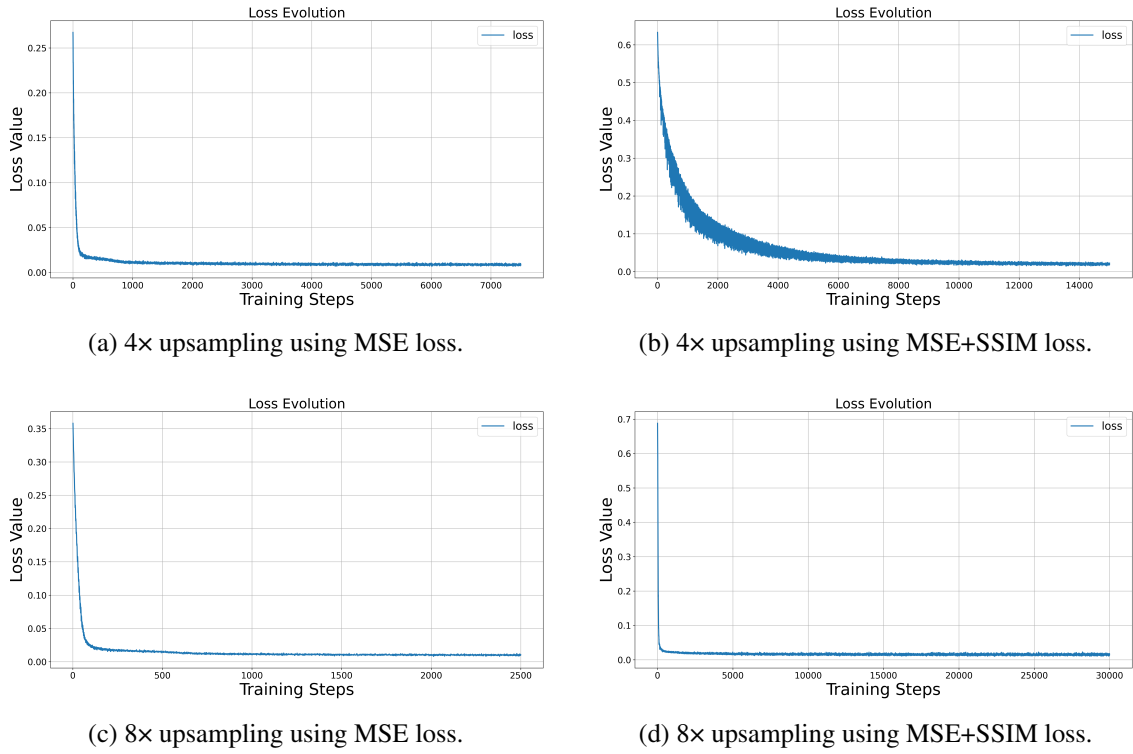


Figure 5.2: Loss evolution curves during training for the baseline experiments. Each curve shows the training loss as a function of training steps for the different configurations.

5.4 Dataset comparison

As the experiments progressed, it became necessary to quantify the actual difference between the LR inputs and their HR ground truth counterparts. This comparison was critical to understanding the model’s performance in context, particularly in assessing whether the model was genuinely learning to enhance input images or simply reproducing content already present in the LR data.

To support this analysis, the evaluation pipeline was revised to numerically compare all the LR and HR versions of the datasets using consistent computation of three key metrics: MSE, SSIM, and GMSE. To compute the metric values for the dataset comparison, each LR sample from the training set was treated as if it were the model’s prediction. The LR and corresponding HR samples were then directly fed into the evaluation metrics, and the resulting scores reflect the differences between them.

The comparison was carried out across the different types of LR datasets: First, it was needed to compare the Real LR dataset with the RGB image representation with the coolwarm colormap of the HR data. Then, the same data, which is represented in single-channel images, will be compared using the grayscale colormap. Then, a final comparison was done between the Real LR data represented with a normalized numerical grid of the LR scalar values in the .npy format and the HR single-channel data.

As shown in Table 5.5, both the RGB and grayscale LR datasets were already very close to



Figure 5.3: 8× Upsampling with Timestep Embedding (MSE + SSIM).

their HR counterparts, with minimal differences across all metrics. The grid-based LR representation showed significantly higher error values.

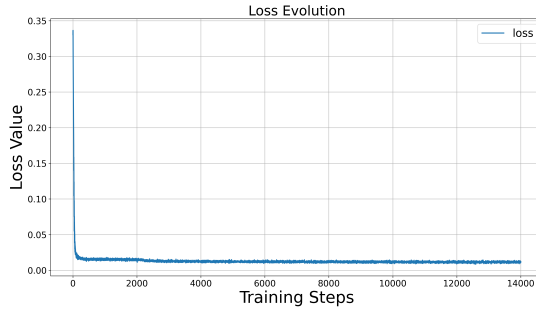
These findings now allow for a more informed interpretation of the results presented in Sections 5.5 and 5.8. Given the minimal difference between the LR and HR datasets, especially in the RGB and grayscale formats, it can be further concluded that most models did not substantially enhance the input data. Instead, they tended to reproduce the already high-quality LR inputs with only minor, often imperceptible modifications.

Table 5.5: Comparison between LR datasets and their HR counterparts

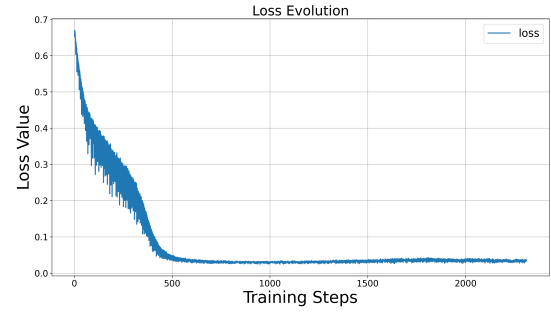
Dataset Comparison	MSE ↓	1-SSIM ↓	GMSE ↓
Real LR (RGB) vs HR	3×10^{-4}	3.7×10^{-3}	4×10^{-4}
Real LR (Grayscale) vs HR	2×10^{-4}	2.7×10^{-3}	3×10^{-4}
lr_mesh_af vs HR	0.0342	0.0317	1.9×10^{-3}

5.5 Real LR Data

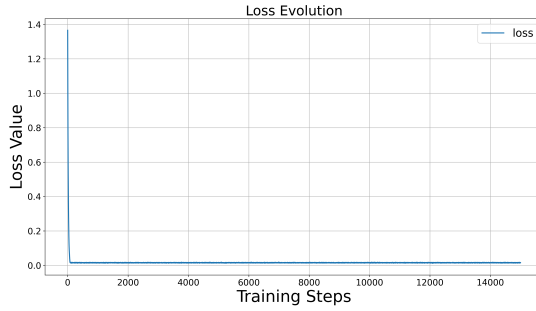
A key direction explored in this work was the use of genuinely LR simulation data obtained from coarser meshes, as opposed to relying exclusively on HR images artificially downsampled via interpolation. The motivation for this was to assess how well the DM generalizes to data distributions that more accurately reflect real-world simulation constraints. In practice, if the available LR data is already quite close to the HR target, it raises the question of whether applying SR brings meaningful improvement. Moreover, relying solely on synthetically degraded HR data for training and evaluation can be misleading, as such degradation may not reflect the characteristics of real LR simulations generated under practical computational constraints. Even when real LR data is available, artificially degraded images remain valuable because real LR datasets are often limited in quantity, diversity, or scope. Synthetic degradation of HR data allows the controlled creation of abundant, diverse, and perfectly aligned LR-HR pairs, which are crucial for supervised training. As mentioned in 4, this required processing simulation outputs stored in .gcns files and converting them into image formats consistent with the HR training data.



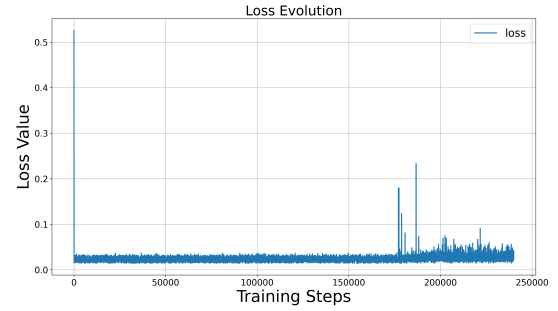
(a) Timestep Embedding using MSE loss.



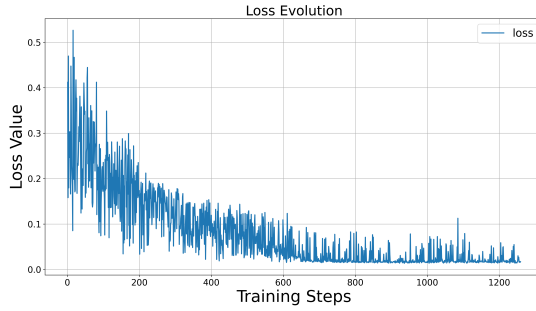
(b) Timestep Embedding using MSE+SSIM loss.



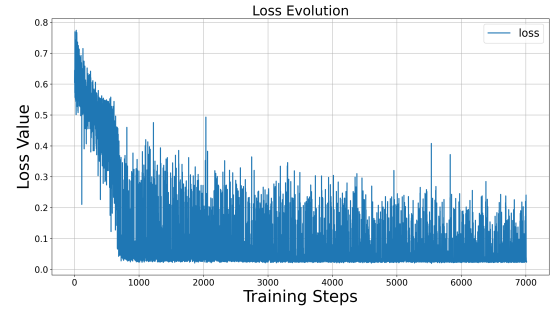
(c) Explicit Conditioning using MSE loss.



(d) Explicit Conditioning using MSE+SSIM loss.



(e) FiLM Conditioning using MSE loss.



(f) FiLM Conditioning using MSE+SSIM loss.

Figure 5.4: Loss evolution during training for the physical conditioning experiments.

To evaluate the model's performance on real LR data. The first involved a simple upsampling using only the MSE as the loss function. The second strategy incorporated timestep conditioning into the model, again using MSE as the loss. The third approach involved an unconditioned upsampling with a hybrid loss function where MSE is combined with SSIM.

The results of these experiments are shown in Table 5.6. All three configurations produced relatively low error values across all evaluated metrics, suggesting that the model is capable of performing effective SR on real LR data. However, the inclusion of timestep conditioning did not improve performance; in fact, it consistently resulted in slightly higher error values. Notably, the combination of MSE and SSIM in the loss function led to the lowest GMSE score, indicating that this approach may better preserve gradient-based structures, even when pixel-wise reconstruction errors remain comparable. As can be observed in 5.4, the LR and HR datasets have minimal

differences between them, especially in the RGB and grayscale formats. This information, coupled with the numerical results obtained by these experiments, leads us to the conclusion that the models did not substantially enhance the input data. An interesting observation is the fact that the model trained to minimize both MSE and SSIM managed to slightly improve the GMSE value of the results.

Qualitative analysis supports these findings but also highlights an important limitation. The predicted outputs closely resemble the HR ground truth, yet the original LR inputs are also already very similar to the ground truth. This makes it difficult to determine whether the DM is meaningfully enhancing the input or simply reproducing it with minimal changes. A visualization example of a result is presented in 5.5.

Table 5.6: Comparison of strategies using real LR data

Strategy	MSE ↓	1-SSIM ↓	GMSE ↓
Upsampling with MSE	5×10^{-4}	7.5×10^{-3}	5×10^{-4}
Timestep Conditioning with MSE	1.6×10^{-3}	0.0183	7×10^{-4}
Upsampling with MSE + SSIM	5×10^{-4}	7.8×10^{-3}	3×10^{-4}

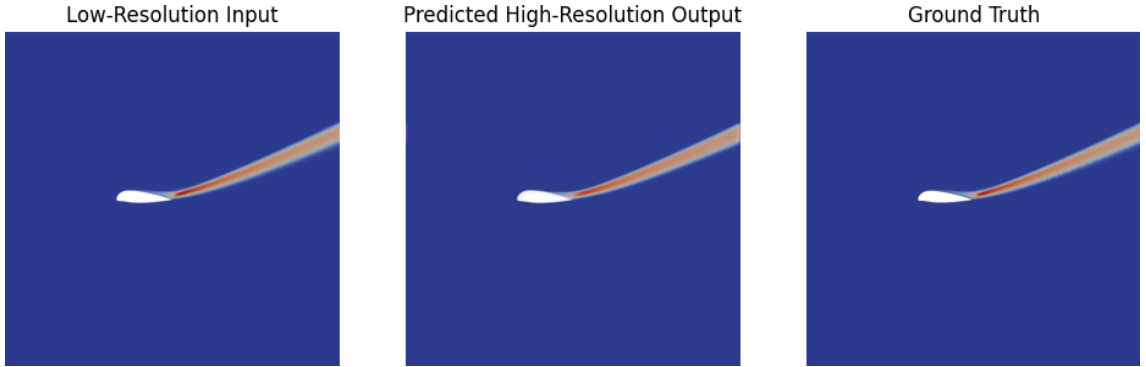


Figure 5.5: 8× Upsampling with real LR input data.

The loss evolution curves in Figure 5.6 show that all models exhibit stable convergence. For the MSE-only and timestep embedding configurations, the loss drops rapidly at the beginning and quickly stabilizes. The model trained with the combined MSE+SSIM loss shows a less abrupt drop and the evolution shows more fluctuations, reflecting the trade-off between minimizing pixel-wise error and preserving structural similarity imposed by the SSIM term.

5.6 Single-channel input

Another line of investigation focused on simplifying the SR task by using single-channel (grayscale) images instead of three-channel RGB inputs. The rationale behind this approach was to reduce input dimensionality while preserving the core physical features of the flow, potentially accelerate convergence, and obtain better results with the same amount of training. To enable this, a new

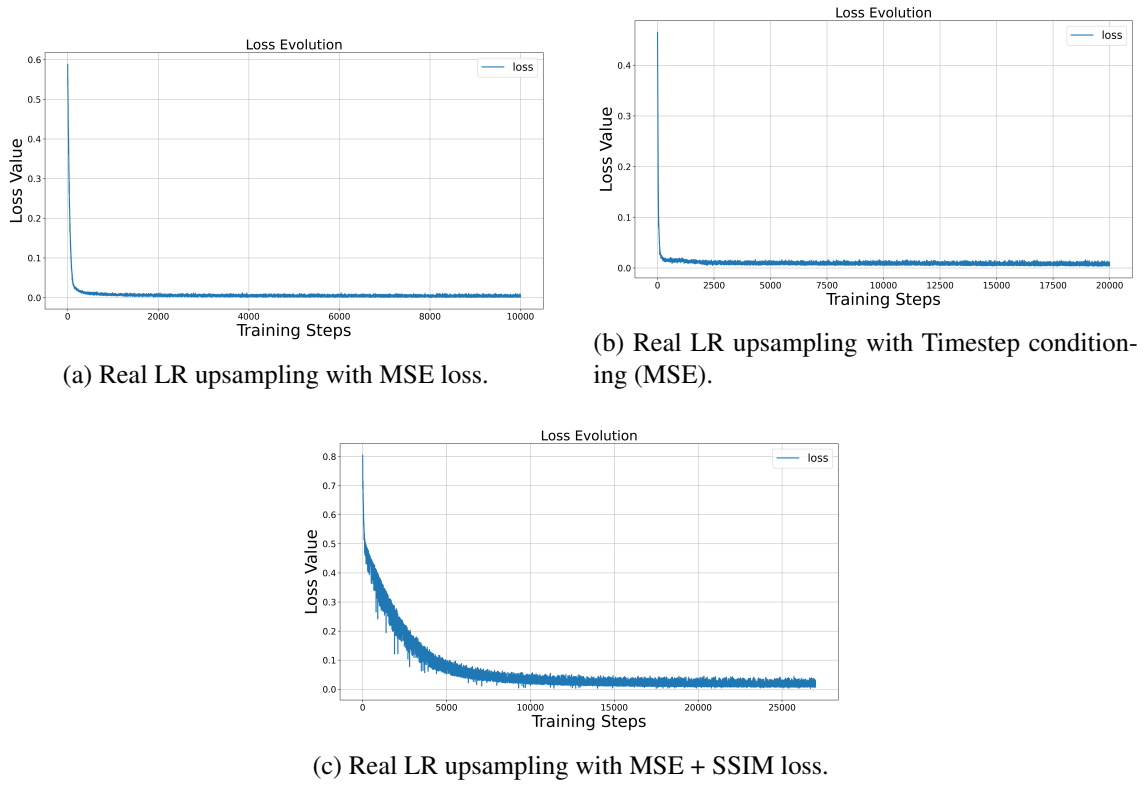


Figure 5.6: Loss evolution curves during training for real LR data experiments.

dataset of grayscale AirfRANS images was generated using an appropriate colormap that maps scalar physical quantities to intensity values; more details are presented in 4.1.

Similar to the 5.5, three experiments were conducted to evaluate model performance under varying conditions. The first involved standard $8\times$ upsampling trained with MSE loss. The second added timestep conditioning to the same $8\times$ setup, also using MSE loss. The third experiment combined MSE with SSIM in the loss function to test whether perceptual guidance could enhance grayscale reconstructions. The results are summarized in Table 5.7.

Notably, the MSE+SSIM loss experiment produced the best performance across all metrics, particularly in terms of structural similarity and gradient preservation. The timestep-conditioned variant, however, underperformed compared to the unconditioned models, showing higher errors in all three metrics. These findings align with earlier results on data with the coolwarm colormap, further suggesting that timestep conditioning may not be beneficial when the input is already close to the target, or when the conditioning signal does not strongly correlate with the residual error.

Despite the theoretical appeal of using single-channel (grayscale) inputs to simplify training, experimental results indicate that the usage of single-channel images does not show significant improvements relative to RGB inputs and shows some slight deterioration in MSE and GMSE results. However, this demonstrates that it can be attributed to a possible richer spatial encoding and higher contrast gradients present in RGB images, which better support learning and improve pixel-level accuracy. In contrast, grayscale conversion compresses this information into a single

intensity range, weakening the signal available to the model and limiting the effectiveness of both standard and perceptual loss functions.

Qualitatively, the results still capture the flow feature very well and closely resemble the HR ground truth image. An example is shown in 5.7.

Table 5.7: Comparison of single-channel (grayscale) strategies

Strategy	MSE ↓	1-SSIM ↓	GMSE ↓
8× upsampling with MSE	1.1×10^{-3}	0.0185	9×10^{-4}
8× upsampling Timestep Conditioned with MSE	2.7×10^{-3}	0.0229	1.2×10^{-3}
8× upsampling with MSE + SSIM	7×10^{-4}	0.0150	7×10^{-4}

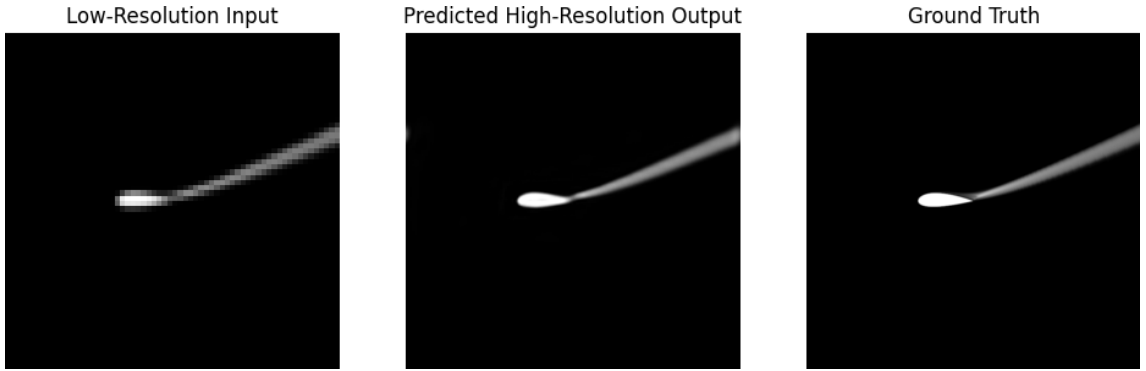


Figure 5.7: 8× unconditioned upsampling with single-channel input.

The loss evolution curves in Figure 5.8 show similar trends across the single-channel experiments, with the loss dropping very quickly and stabilizing at low values in all cases. Even the model trained with MSE+SSIM converges efficiently and with less fluctuations, unlike in the real LR experiments where it showed slower and noisier convergence. This suggests that using single-channel input helps the model learn more effectively.

5.7 GMSE

In addition to architectural and data-based conditioning strategies, we also explored an alternative approach: conditioning the model through the loss function itself. Specifically, we incorporated the already studied GMSE metric, introduced in [15]. GMSE tries to achieve better structural accuracy by emphasizing regions of the image with high spatial variance and dynamically assigning greater weight to gradient-rich areas during training. The motivation for incorporating GMSE was the possibility of the model better capturing flow features and sharp transitions, an essential requirement in this context, where RANS simulations often exhibit distinct and well-defined flow structures.

To evaluate the effectiveness of this approach, two configurations were tested using grayscale inputs to reduce training complexity and speed up experimentation. The first involved using

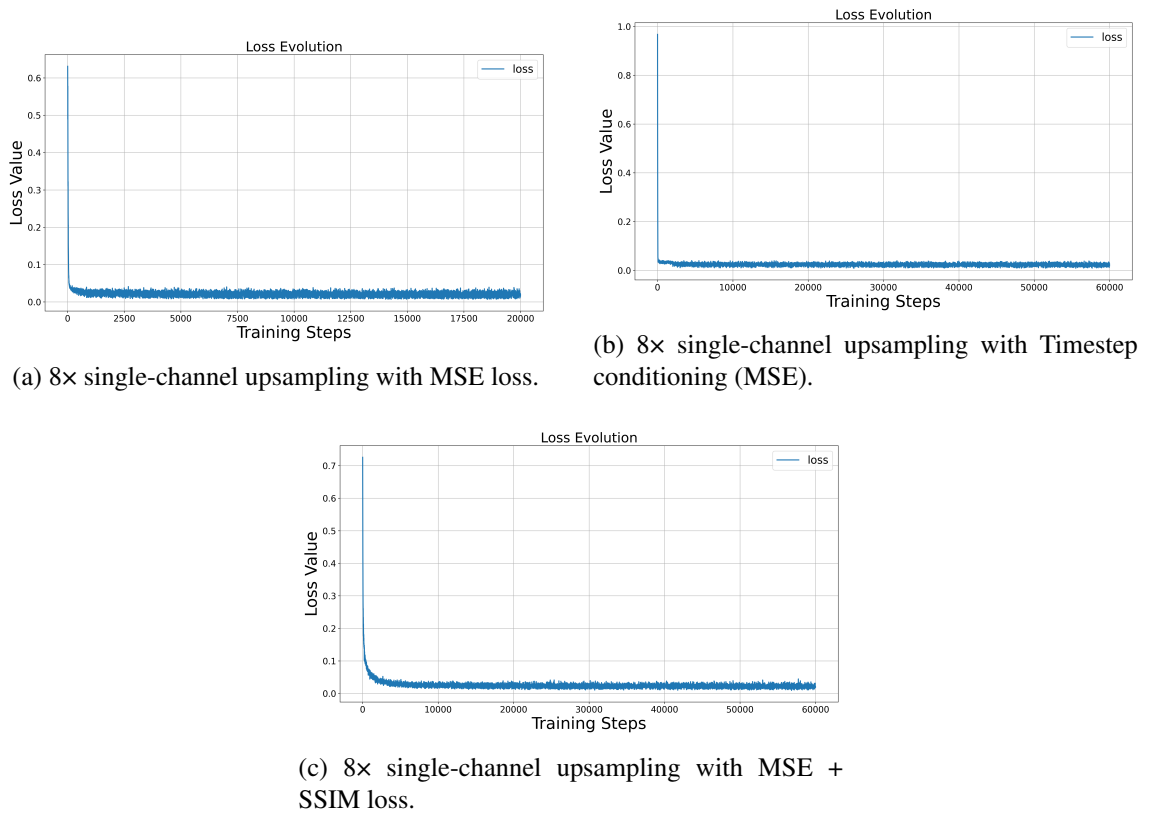


Figure 5.8: Loss evolution curves during training for single-channel input experiments.

GMSE as the sole loss function, with no contribution from MSE. The second combined GMSE with MSE in a timestep-conditioned training setup. The results are presented in Table 5.8.

Interestingly, the model trained using GMSE alone yielded a very low GMSE score but extremely high MSE and SSIM errors, indicating that while gradients were preserved, the overall image structure and intensity values diverged significantly from the ground truth. This suggests that optimizing exclusively for GMSE may cause the model to ignore global consistency in favor of local edge alignment. On the other hand, the hybrid loss configuration (MSE + GMSE with timestep conditioning) achieved more balanced results across all metrics, maintaining reasonable gradient accuracy while substantially reducing pixel-wise and structural errors. This confirms that while GMSE can be a valuable addition to the loss function, it is best used in conjunction with more traditional objectives like MSE.

Visually, the GMSE-only model results demonstrate strong capability in recovering sharp edges and flow features, such as boundary layers. An example visualization of a result is present in 5.9. However, the absence of pixel-wise guidance from MSE leads to significant global inconsistencies and many minor imperfections across the background, which are likely the cause behind the high MSE. This reinforces the idea that while GMSE is effective at preserving local gradient structures, the lack of combination with pixel-wise metrics causes inaccurate global reconstruction. The visual artifacts and over-emphasized contrasts seen in the prediction highlight the model’s tendency to overfit to gradient cues when not regularized by broader error metrics.

Table 5.8: Comparison of strategies using GMSE in the loss function

Strategy	MSE ↓	1-SSIM ↓	GMSE ↓
GMSE-only (no MSE)	0.6771	0.6677	2×10^{-4}
Timestep Conditioning with MSE + GMSE	2.8×10^{-3}	0.0230	1.2×10^{-3}

The loss evolution curves in Figure 5.10 illustrate the training dynamics of the GMSE experiments. For the model trained solely with GMSE, the loss drops quickly but stabilizes at a much higher level compared to other experiments, reflecting its poor performance in pixel-wise and structural metrics despite its low gradient error. In contrast, the model trained with the MSE+GMSE loss function and conditioned by the timestep exhibits lower overall loss values, confirming that combining GMSE with traditional objectives like MSE benefits the model without the trade-off issues observed in the MSE+SSIM experiments.

5.8 Advanced Physical conditionings

The initial physical conditioning strategy, referred to as Explicit Conditioning as Additional Channel (`cond_exp`), consisted of adding two constant-valued channels to the model input: one encoding the Reynolds number and the other the angle of attack. These values were repeated across every pixel of the input image. While this approach aimed to introduce a basic form of physical awareness, the results were ultimately underwhelming. However, it laid a useful foundation for embedding physical simulation parameters directly into the learning pipeline.

With the goal to further explore strategies to improve the results of DMs with access to physical parameters, two more advanced strategies were developed: `exp_imp` and `exp_imp_full`. Both are designed to take the single-channel version of each sample and inject richer physical context into the model by expanding the number and type of conditioning channels.

The first approach, `exp_imp`, adds up to four new input channels, one channel was added for each of the following information: Reynolds number, angle of attack, and the mean values of velocity, pressure, and viscosity. To obtain the mean value of each scalar value, pressure, velocity, and viscosity, for each simulation sample, these values are averaged over the image. Then the new channels will have this mean value repeated across each pixel.

An important problem to take into account is data leakage, where the model could obtain revealing information about the target output from the input. To prevent any kind of leakage, the channel with the mean value of the scalar being predicted is excluded from the input. For example, if the model is trained to predict viscosity, the mean viscosity channel is omitted. This ensures that only non-redundant, complementary information is used to guide the model.

The second method, `exp_imp_full`, also adds four channels, but instead of using global averages, it incorporates the full spatial fields of the physical variables. This means that the additional channels include the actual per-pixel distributions of velocity, pressure, and viscosity, which are drawn directly from the dataset. As with the previous method, the predicted target variable is excluded; in this case, the viscosity channel was not added. This approach gives the model access



Figure 5.9: 8× unconditioned upsampling with GMSE-only loss function.

to localized physical features and complex spatial variations of related scalar values, which can definitely be beneficial to the DM when modeling flow phenomena with strong spatial dependencies.

Both conditioning strategies are applied in the context of LR grayscale input images (`real_lr`) and require additional pre-processing steps to compute or extract the relevant physical quantities and grids.

The evaluation metrics of the results of these experiments are presented in 5.9. Quantitatively, one noticeable finding is the significant improvement of the advanced conditioning strategies compared to the timestep embedding approach applied to the same dataset, as described in 5.7. This suggests that the DM was able to extract useful information from the additional physical inputs and leverage it to produce more accurate results. Still, given the minimal differences observed between the LR and HR datasets as described in 5.4, it can also be concluded that these models did not meaningfully enhance the input data.

A qualitative analysis of the results is also hard to do, due to the similarity of both HR and LR data, but the predicted images show a visual solid resemblance to the HR image. A result example of `exp_imp` can be found in 5.11, and for `exp_imp_full` can be found in 5.12. The input and output similarity alludes to a possible limitation where even the minimal deviation makes it difficult to assess whether the improvements are physically meaningful or simply reflect small, possibly uninformative, adjustments. In such cases, even low error metrics may not correspond to genuine improvements in predictive quality, emphasizing the need for careful interpretation of the results and numerical comparison of the LR and HR datasets.

Table 5.9: Comparison of advanced physical conditioning strategies

Strategy	MSE ↓	1-SSIM ↓	GMSE ↓
<code>exp_imp</code>	0.0349	0.0324	2×10^{-3}
<code>exp_imp_full</code>	9×10^{-4}	0.0154	7×10^{-4}

The loss evolution curves for the advanced conditioning strategies are shown in Figure 5.13. The `exp_imp` configuration appears erratic, largely due to the scale of the plot, but the loss itself remains almost unchanged throughout training, indicating ineffective learning. The `exp_imp_full`

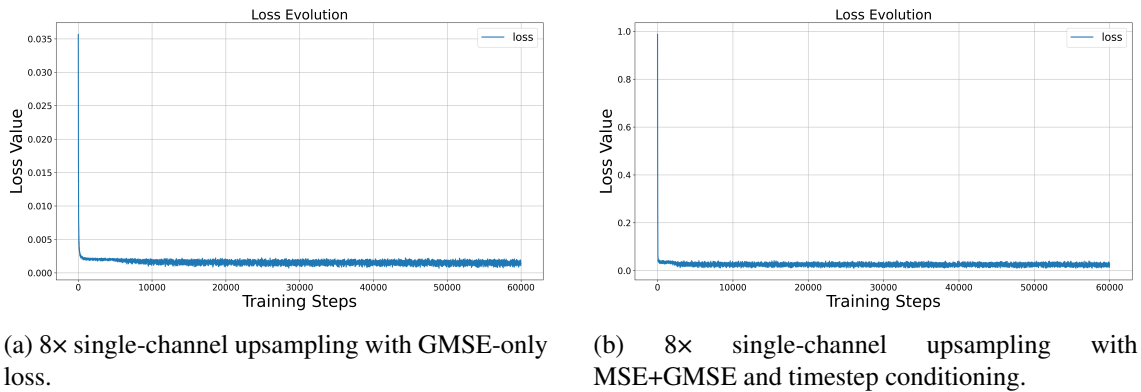


Figure 5.10: Loss evolution curves during training for the GMSE experiments.

configuration exhibits a seemingly stable loss curve, aside from a single outlier spike. Ignoring this outlier, the loss remains low and does not show substantial improvement during training. This suggests that the model does not effectively minimize the loss further; the detailed spatial fields provided as conditioning in `exp_imp_full` allow it to achieve good performance despite the lack of significant convergence.

5.9 Transfer Learning

As another interesting experiment to further evaluate whether the model is truly learning meaningful features related to fluid flow patterns, a new experiment was conducted involving the application of transfer learning. The goal was to assess whether a DM, pre-trained on the standard `airFRANS` dataset, could generalize to a slightly different spatial representation of the same underlying physical fields.

The core idea behind this experiment is to reuse the latent representations learned by the DM by freezing the layers theoretically responsible for that task. Then, retraining the remaining layers, responsible for reconstructing the final image from the extracted latent features, on a new data distribution. The experiment introduces a transformed version of the input data presented in 4.1.4.

In implementing transfer learning, the majority of the model's parameters and layers, particularly in the encoder and middle blocks, are frozen to retain the general features already learned. Only the final layers of the decoder path, which are responsible for decoding and reconstructing the output, are fine-tuned on the new data. This setup assumes that the model has already learned to capture the essential dynamics of the flow in its latent space, and only the final mappings need to adapt to the new spatial structure.

For this specific experiment, two models were tested under this scheme, each corresponding to a previously trained configuration: `airfrans_run_real_lr_cond_exp_imp_mse` and `airfrans_run_real_lr_cond_exp_imp_full_mse`. The results are shown in Table 5.10.

As shown in Table 5.10, both transfer learning configurations yielded higher errors than the

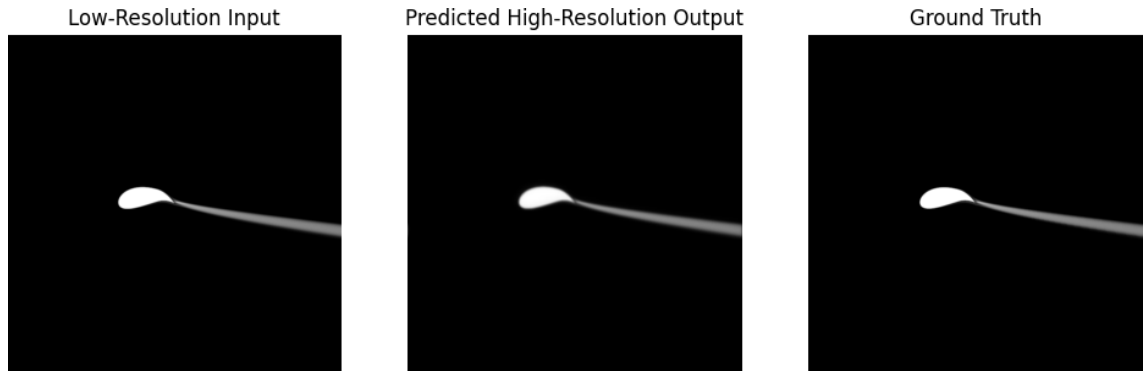


Figure 5.11: Visualization example of a exp_imp result.

baseline, which here refers to the comparison between the LR and HR versions of the corresponding datasets. However, their relative performance improved notably with transfer learning. The percentage increase in MSE was computed as:

$$Relative_increase(\%) = \frac{MSE_experiment - MSE_baseline}{MSE_baseline} \times 100$$

For exp_imp_mse, the MSE was approximately 17,350% higher than the HR baseline, but dropped to only 2.05% higher with transfer learning. For exp_imp_full, the increase dropped from 350% (HR baseline) to 41% with transfer learning. This indicates that the transfer learning technique actually helped the models achieve results closer to their HR counterparts. Interestingly, while exp_imp_full outperformed exp_imp_mse in the non-transfer experiments, its transfer learning variant performed slightly worse than exp_imp_mse with transfer learning. This suggests that transfer learning indeed improved both strategies significantly by leveraging latent features from the pre-trained models, but the benefit was more pronounced for exp_imp_mse.

This is in contrast with a qualitative analysis of the result images, which seems to show a huge degradation in performance in both experiments. As is shown in Figure 5.14 and Figure 5.15.

Table 5.10: Comparison of experiments applying transfer learning

Type	MSE ↓	SSIM ↓	GMSE ↓
exp_imp_mse_transfer_learning	0.0349	0.0323	2×10^{-3}
exp_imp_full_mse_transfer_learning	0.0482	0.0659	2.3×10^{-3}

The loss evolution curves for the transfer learning experiments are shown in Figure 5.16. For the exp_imp transfer learning configuration, the loss remains largely and relatively flat throughout training, with no significant decrease, indicating limited adaptation to the new data distribution and suggesting that the frozen latent representations did not effectively capture the required adjustments. In contrast, the exp_imp_full configuration exhibits a more structured and gradually decreasing loss curve, reflecting better learning dynamics.

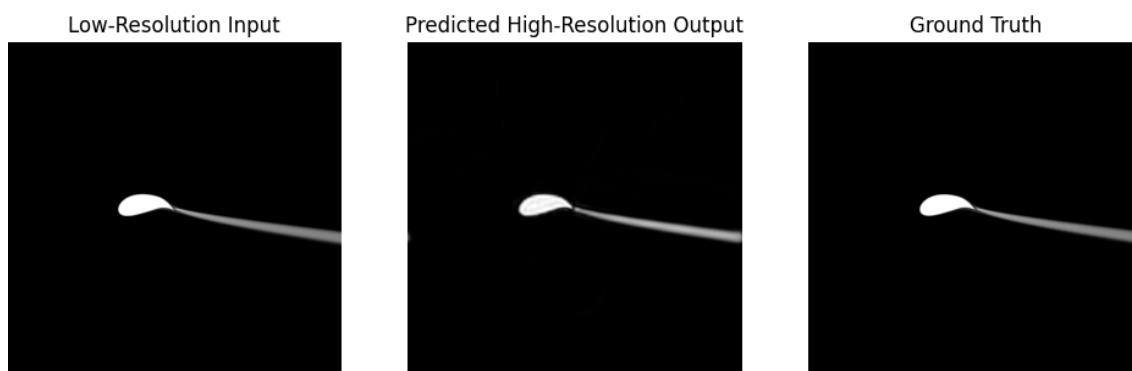


Figure 5.12: Visualization example of a `exp_imp_full` result.

5.10 Discussion

The experiments conducted in this chapter highlight both the capabilities and limitations of DMs for SR tasks in CFD, airfRANS data. The best results were achieved by baseline models trained on synthetically downsampled HR images, which consistently produced low error values and visually accurate and stronger reconstructions. When using single-channel inputs, performance remained comparable, but no consistent improvement was observed—likely due to the loss of spatial richness inherent to RGB representations.

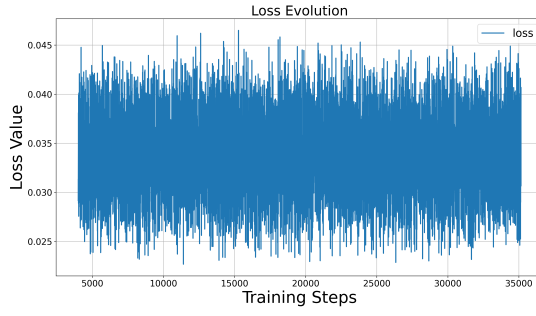
Simple physical conditioning strategies yielded limited success. While timestep embedding consistently outperformed FiLM and explicit conditioning, none of these approaches surpassed the unconditioned baseline. In particular, FiLM and explicit conditioning often led to overfitting, with the models focusing excessively on background color patterns rather than capturing meaningful flow dynamics.

Loss function design played a significant role in model performance. MSE provided a reliable baseline, SSIM occasionally improved structural coherence, and GMSE proved effective in enhancing gradient features. However, GMSE alone led to poor global accuracy, and its benefits were only realized when combined with MSE.

Experiments with real LR simulation data showed that DMs can reproduce HR outputs, but with minimal improvement over the input. This is largely due to the already high fidelity of the LR data, which makes it difficult to discern whether the model is truly enhancing the input or simply replicating it.

More advanced conditioning approaches, such as `exp_imp_full`, demonstrated better performance than simple physical conditioning, likely owing to the inclusion of full spatial physical fields. Still, the very small differences between input and ground truth limited the interpretability of improvements and underscored the importance of using datasets that challenge the model more meaningfully.

Transfer learning experiments on real LR data still did not produce better results than simply replicating the LR. However, one can argue that knowledge reuse led to a slight improvement in performance relative to the base case, which is the comparison between the LR grid data and the



(a) exp_imp advanced conditioning.



(b) exp_imp_full advanced conditioning.

Figure 5.13: Loss evolution curves during training for the advanced physical conditioning experiments.

HR data. The qualitative results show a prediction that is very far from either the LR input or the actual HR ground truth.

Overall, DM demonstrates strong potential for SR in CFD applications. However, their effectiveness is closely tied to the design of conditioning strategies, the choice of loss functions, and the complexity of the input data representations. As discussed in Chapter 3, DMs have seen more extensive application in highly turbulent flow scenarios, suggesting that their strengths are best leveraged in contexts with greater variability and richer physical structures. This may represent the most suitable and impactful use case for DMs in fluid dynamics.

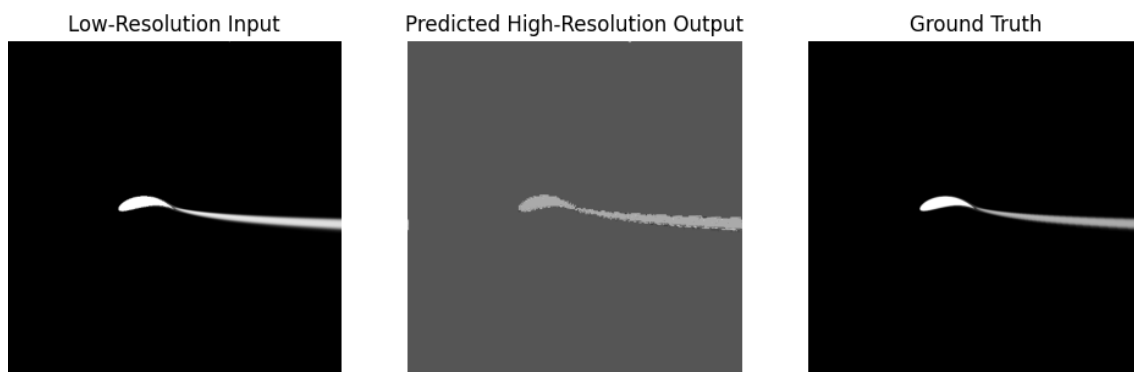


Figure 5.14: Visualization example of a `exp_imp` result with transfer learning.

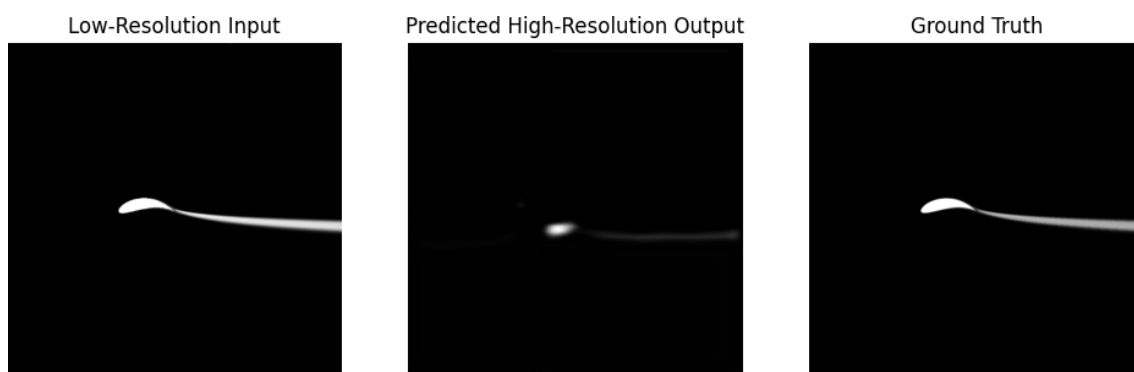
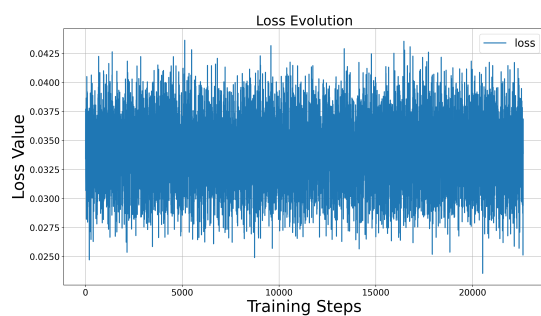
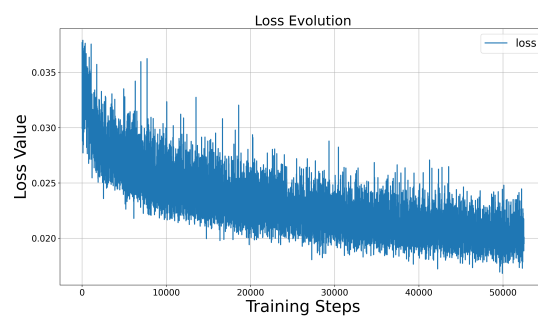


Figure 5.15: Visualization example of a `exp_imp_full` result with transfer learning.



(a) `exp_imp` transfer learning.



(b) `exp_imp_full` transfer learning.

Figure 5.16: Loss evolution curves during training for the advanced conditioning strategies with transfer learning.

Chapter 6

Conclusion

This chapter presents a synthesis of the core objectives, findings, and contributions of this dissertation. It begins by revisiting the main motivations behind the research, summarizing what was achieved both theoretically and experimentally. The discussion then addresses the central research questions posed at the beginning of the work, analyzing how each was answered and highlighting the most relevant results obtained throughout the study.

Following this, the chapter outlines potential future developments. These include both short-term to medium-term directions, focused on continuing the current research trajectory, as well as longer-term opportunities that aim to deepen and broaden the understanding and applicability of DMs in CFD. Topics such as explainability, robustness, stress testing, and alternative model architectures are briefly discussed to indicate fruitful paths for future exploration.

6.1 Main Contributions

The central aim of this dissertation was to evaluate the applicability and effectiveness of DMs for SR tasks in the context of CFD, using the airFRANS dataset as a representative benchmark.

This project began with extensive research through a literature review that established the theoretical and practical foundations for this work. It covered the principles of generative modeling, including GANs, DMs, and other classical surrogates like POD and DL-based, highlighting their application in fluid simulation contexts. Advances in PIML were explored, with special attention to physical conditioning strategies that embed domain knowledge, such as governing equations, boundary constraints, or others, into neural architectures or loss functions. The review also examined a wide variety of datasets used in CFD-related research, from canonical cases like Kolmogorov flows and airfoil wake simulations to complex real-world configurations. Finally, a wide range of evaluation metrics was surveyed, including point-wise metrics, such as MSE, perceptual metrics, such as SSIM and LPIPS, and physically meaningful measures, emphasizing the importance of selecting appropriate criteria to ensure numerical accuracy, human-level visual fidelity, and physical validity in the model’s predictions. Finally, there’s an exploration of transfer learning techniques, focusing on their potential to improve generalization and training efficiency by

leveraging knowledge from pre-trained models across similar fluid dynamics problems or configurations. This includes domain adaptation techniques and the reuse of learned representations in new simulation settings.

The main contribution of this dissertation lies in the development of a comprehensive set of experiments conducted to explore various model configurations, focusing on differences in input data representations and data types, such as different amounts of channels and different fidelities of LR data, loss functions, and physical conditioning strategies. These configurations were tested to assess the models' ability to generalize and produce high-fidelity outputs.

Each experiment addressed several critical dimensions of SR for CFD, including the integration of physical parameters into generative models, the role of perceptual and gradient-based loss metrics, and the potential for knowledge transfer through pretraining. The results of each experiment were quantitatively and qualitatively analyzed, and the relevant conclusions were also explained and founded.

Bringing back the research questions posed to be answered by this dissertation in Chapter 1.4, it is now possible to reflect upon how this dissertation provided the necessary foundation to answer them.

Can DMs be used to accurately upsample airfRANS simulations?

The experimental results demonstrate that DMs can be highly effective for SR tasks when applied to airfRANS data, but only when inserted in scenarios involving heavily synthetically downsampled inputs. Models trained under these conditions consistently achieved low error rates across key studied metrics (MSE, SSIM, and GMSE), and visual analysis confirmed that the outputs accurately reconstructed the main flow features, such as airfoil contours and viscosity gradients.

However, when evaluated on real LR simulation data, where the input is already visually and numerically very close to the HR ground truth, the performance gains diminished and were quite negligible. In these cases, the models closely reproduced the input with only minor refinements, but this offered degradations in the actual results. This suggests that the strength of DMs lies in bridging significant resolution gaps, whereas their impact is less pronounced when the initial input is already high-quality.

How to condition generative models with physical laws and parameters to ensure that SR tasks on fluid flow simulation data produce realistic results that really represent the phenomena under study?

The experiments conducted in this dissertation reveal that conditioning generative models with physical information, particularly the initial conditions of the simulation, can enhance the model's ability to produce more accurate HR outputs. This conditioning provides additional context that helps the model infer physically plausible reconstructions. However, this benefit comes with notable drawbacks. The inclusion of physically meaningful inputs increases the dimensionality of the data and introduces additional dependencies, which in turn make the training process more

computationally demanding and potentially less stable. These effects highlight the importance of carefully considering the trade-offs of this technique and introduce ways to avoid converging to local optima.

To explore this further, an advanced conditioning technique was developed in which the model's inputs were augmented with additional channels representing key physical quantities. These included either the full spatial fields or spatially averaged values of scalar quantities such as pressure and velocity magnitude. By explicitly embedding this information into the model's input space, the network could more easily learn correlations between LR structures and their HR counterparts, guided by the physical state of the system. This conditioning approach improved the model's performance in reconstructing fine-scale features and contributed to greater physical realism in the predictions when compared to simpler physical conditioning techniques. These findings suggest that while physical conditioning can be a valuable asset for SR in CFD, its effectiveness depends on the specific nature of the task and must be integrated thoughtfully.

What are the challenges and the differences between using artificial downsampled HR simulations as LR and real LR simulations?

One of the key challenges uncovered in this dissertation relates to the discrepancy between artificially downsampled HR simulations and genuinely LR simulation data. Artificial LR data, typically generated by applying interpolation techniques to HR outputs, significantly reduces the fidelity of the LR way more than actual realistic LR simulations, and in a consistent manner. As a result, models trained and evaluated on such data often exhibit optimistic performance, as the mapping between LR and HR becomes a relatively straightforward inversion of a known and consistent transformation.

The realistically obtained LR inputs are significantly less noisy and closely aligned with their corresponding HR outputs. In these cases, the LR data is already very similar to the HR target, meaning that even simple replication of the input yields good results. This makes it challenging for the model to identify and learn the subtle differences required to meaningfully improve the prediction, as the margin for enhancement is minimal and the signal guiding the upsampling is weak.

Does GMSE in the loss function improve CFD predictions?

The inclusion of GMSE in the loss function was found to be beneficial for enhancing the model's sensitivity to sharp transitions, which are common in RANS simulations, such as boundary layers and shear regions. By penalizing differences in spatial gradients rather than just absolute values, GMSE encourages the model to preserve structural details that are often critical in CFD applications but may be overlooked by traditional MSE-based training alone.

However, experiments show that GMSE on its own does not consistently lead to better overall performance. When used without complementary losses like MSE, it may fail to enforce global consistency, resulting in visually sharper but less accurate reconstructions in terms of physical

fields. The best results were obtained when GMSE was combined with MSE, allowing the model to balance both detailed structure and general field accuracy. This suggests that while GMSE is a valuable component for improving CFD predictions, it should be integrated as part of a multi-term loss function rather than relied upon in isolation.

Can transfer learning help the model achieve better or faster results?

From the transfer learning experiments, it is possible to argue that the model's results were closer to their target HR dataset than the homologous non-transfer learning version. However, neither experiment managed to improve the real LR version of the data relative to the HR data. And from the qualitative analysis of the results, we can verify a higher degradation of the results. These mixed conclusions do not allow an affirmative statement regarding transfer learning improving DM's performance using this particular setup. An important point to make is that the different data domains were probably too far apart for the possibility of transferring knowledge between them. Transfer learning is still a promising strategy worth researching in this context, but it will need other types of, possibly more complex, techniques, like a variation on the ones presented at [45, 77].

6.2 Further Developments

Building upon the findings of this dissertation, there are several promising directions for further development, both in the short to medium term and in longer-term research efforts.

In the short to medium term (3 months to 1 year), future work could focus on refining the current experimental setup and addressing some of its identified limitations. This includes exploring simpler and more effective strategies for physically conditioning the DM, as well as optimizing the loss function by identifying an ideal balance between MSE and GMSE. Particular attention should be given to how this balance affects its alignment with the evaluation metrics used to assess performance.

Additionally, increasing the diversity of the training datasets could significantly improve the model's generalization capabilities, particularly when applied to more complex or previously unseen flow scenarios. This involves evaluating the DM on a broader set of fluid dynamics problems, such as bubbly flows, Rayleigh–Bénard convection, and flow past a cylinder. These cases span a variety of physical regimes and flow structures, providing a more rigorous assessment of the model's robustness and adaptability. Moreover, such experiments would help validate the hypothesis that DMs tend to perform better on more turbulent or highly dynamic flows, where their generative capacity can more effectively reconstruct complex spatial features. Another important step would be to validate the models further using real LR simulation data from different solvers or mesh configurations to assess their robustness and practical applicability in production environments.

Also, within the short-term goals, a deeper exploration of transfer learning strategies, particularly with carefully selected pretraining domains, could enhance the efficiency and adaptability of the models.

Looking further ahead (1 to 3–5 years), the research opens avenues for investigating more fundamental aspects of DL applied to CFD. One such direction is the development of methods for model explainability, enabling a better understanding of how the generative models make predictions and how they respond to changes in input conditions. Another is stress testing the models across a wide range of flow regimes and physical edge cases to assess their limits and identify potential failure modes. Robustness analysis, especially under perturbations in input data or variations in physical parameters, is also crucial for ensuring reliability in real-world deployments. Finally, exploring alternative model architectures, such as the Kolmogorov–Arnold Network (KAN), which offers a new way of representing complex functions, could significantly improve the expressiveness and interpretability of surrogate models in CFD tasks.

Together, these developments aim to strengthen the physical fidelity, reliability, and interpretability of generative approaches for fluid simulation, ultimately contributing to more robust and scalable AI-driven tools in scientific and engineering domains.

References

- [1] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. arXiv: 1603.04467 [cs.DC]. URL: <https://arxiv.org/abs/1603.04467>.
- [2] Fraunhofer AICOS. *pyMDMA: Multimodal Data Metrics for Auditing real and synthetic datasets*. 2024. URL: <https://github.com/fraunhoferportugal/pymdma>.
- [3] M. J. Asher et al. “A review of surrogate models and their application to groundwater modeling”. In: *Water Resources Research* 51.8 (2015), pp. 5957–5973. DOI: <https://doi.org/10.1002/2015WR016967>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2015WR016967>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2015WR016967>.
- [4] Romain Aubry et al. “An Unstructured CFD Approach for Numerical Weather Prediction”. In: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. DOI: 10.2514/6.2010-691. eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2010-691>. URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2010-691>.
- [5] Chayan Banerjee et al. “Physics-Informed Computer Vision: A Review and Perspectives”. In: *ACM Comput. Surv.* (2024). arXiv: 2305.18035 [eess.IV]. URL: <https://arxiv.org/abs/2305.18035>.
- [6] Manuel Berger et al. “A comparison between 2D DeepCFD, 2D CFD simulations and 2D/2C PIV measurements of NACA 0012 and NACA 6412 airfoils”. In: *Engineering Science and Technology, an International Journal* 57 (2024), p. 101794. ISSN: 2215-0986. DOI: <https://doi.org/10.1016/j.jestch.2024.101794>. URL: <https://www.sciencedirect.com/science/article/pii/S2215098624001800>.
- [7] A. Bonanni et al. “Wind farm optimization based on CFD model of single wind turbine wake”. In: *European Wind Energy Conference and Exhibition 2012, EWEC 2012 2* (Jan. 2012), pp. 726–735.
- [8] Florent Bonnet et al. *AirfRANS: High Fidelity Computational Fluid Dynamics Dataset for Approximating Reynolds-Averaged Navier-Stokes Solutions*. 2023. arXiv: 2212.07564 [cs.LG]. URL: <https://arxiv.org/abs/2212.07564>.
- [9] Pierre-Emmanuel Bournet and Fernando Rojano. “Advances of Computational Fluid Dynamics (CFD) applications in agricultural building modelling: Research, applications and challenges”. In: *Computers and Electronics in Agriculture* 201 (2022), p. 107277. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2022.107277>. URL: <https://www.sciencedirect.com/science/article/pii/S0168169922005890>.

- [10] Shengze Cai et al. *Physics-informed neural networks (PINNs) for fluid mechanics: A review*. 2021. arXiv: 2105.09506 [physics.flu-dyn]. URL: <https://arxiv.org/abs/2105.09506>.
- [11] Giovanni Catalani et al. “A comparative study of learning techniques for the compressible aerodynamics over a transonic RAE2822 airfoil”. In: *Computers and Fluids* 251 (2023). DOI: 10.1016/j.compfluid.2022.105759. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85143851812&doi=10.1016%2fj.compfluid.2022.105759&partnerID=40&md5=dc42eb307bae354a982fac493632e3b1>.
- [12] Honggang Chen et al. “Real-world single image super-resolution: A brief review”. In: *Information Fusion* 79 (2022), pp. 124–145. ISSN: 1566-2535. DOI: <https://doi.org/10.1016/j.inffus.2021.09.005>. URL: <https://www.sciencedirect.com/science/article/pii/S1566253521001792>.
- [13] Mengyu Cheng et al. “A Review of Data-Driven Surrogate Models for Design Optimization of Electric Motors”. In: *IEEE Transactions on Transportation Electrification* 10.4 (2024), pp. 8413–8431. DOI: 10.1109/TTE.2024.3366417.
- [14] Xuesen Chu et al. “Flow reconstruction over a SUBOFF model based on LBM-generated data and physics-informed neural networks”. In: *Ocean Engineering* 308 (2024), p. 118250. ISSN: 0029-8018. DOI: <https://doi.org/10.1016/j.oceaneng.2024.118250>. URL: <https://www.sciencedirect.com/science/article/pii/S0029801824015889>.
- [15] Zachary Cooper-Baldock et al. *A generalised novel loss function for computational fluid dynamics*. 2024. arXiv: 2411.17059 [cs.LG]. URL: <https://arxiv.org/abs/2411.17059>.
- [16] Prafulla Dhariwal and Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*. 2021. arXiv: 2105.05233 [cs.LG]. URL: <https://arxiv.org/abs/2105.05233>.
- [17] Pan Du et al. *CoNFILd: Conditional Neural Field Latent Diffusion Model Generating Spatiotemporal Turbulence*. 2024. arXiv: 2403.05940 [physics.flu-dyn]. URL: <https://arxiv.org/abs/2403.05940>.
- [18] Wael Emad et al. “Prediction of concrete materials compressive strength using surrogate models”. In: *Structures* 46 (2022), pp. 1243–1267. ISSN: 2352-0124. DOI: <https://doi.org/10.1016/j.istruc.2022.11.002>. URL: <https://www.sciencedirect.com/science/article/pii/S2352012422010414>.
- [19] Xiantao Fan, Deepak Akhare, and Jian-Xun Wang. *Neural Differentiable Modeling with Diffusion-Based Super-resolution for Two-Dimensional Spatiotemporal Turbulence*. 2024. arXiv: 2406.20047 [physics.flu-dyn]. URL: <https://arxiv.org/abs/2406.20047>.
- [20] Edward Ferdian et al. “4DFlowNet: Super-Resolution 4D Flow MRI Using Deep Learning and Computational Fluid Dynamics”. In: *Frontiers in Physics* 8 (May 2020). ISSN: 2296424X. DOI: 10.3389/fphy.2020.00138.
- [21] 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”. In: *Physics of Fluids* 33.7 (July 2021), p. 073603. ISSN: 1089-7666. DOI: 10.1063/5.0054312. URL: <http://dx.doi.org/10.1063/5.0054312>.
- [22] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

- [23] Shuyang Gu et al. *Vector Quantized Diffusion Model for Text-to-Image Synthesis*. 2022. arXiv: 2111.14822 [cs.CV]. URL: <https://arxiv.org/abs/2111.14822>.
- [24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG]. URL: <https://arxiv.org/abs/2006.11239>.
- [25] Jonathan Ho et al. *Cascaded Diffusion Models for High Fidelity Image Generation*. 2021. arXiv: 2106.15282 [cs.CV]. URL: <https://arxiv.org/abs/2106.15282>.
- [26] Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. “A Review of Deep Transfer Learning and Recent Advancements”. In: *Technologies* 11.2 (2023). ISSN: 2227-7080. DOI: 10.3390/technologies11020040. URL: <https://www.mdpi.com/2227-7080/11/2/40>.
- [27] Mehmet Kaya, Ali Kök, and Huseyin Kurt. “Comparison of Aerodynamic Performances of Various Airfoils from Different Airfoil Families Using CFD”. In: *Wind and Structures An International Journal* 32 (Mar. 2021), pp. 239–248. DOI: 10.12989/was.2021.32.3.239.
- [28] Rinil Kuriakose and C. Anandharamakrishnan. “Computational fluid dynamics (CFD) applications in spray drying of food products”. In: *Trends in Food Science and Technology* 21.8 (2010), pp. 383–398. ISSN: 0924-2244. DOI: <https://doi.org/10.1016/j.tifs.2010.04.009>. URL: <https://www.sciencedirect.com/science/article/pii/S0924224410001263>.
- [29] Dawa Chyophel Lepcha et al. “Image super-resolution: A comprehensive review, recent trends, challenges and applications”. In: *Information Fusion* 91 (2023), pp. 230–260. ISSN: 1566-2535. DOI: <https://doi.org/10.1016/j.inffus.2022.10.007>. URL: <https://www.sciencedirect.com/science/article/pii/S1566253522001762>.
- [30] Matthew Li and Christopher McComb. “Using Physics-Informed Generative Adversarial Networks to Perform Super-Resolution for Multiphase Fluid Simulations”. In: *Journal of Computing and Information Science in Engineering* 22.4 (Feb. 2022), p. 044501. ISSN: 1530-9827. DOI: 10.1115/1.4053671. eprint: https://asmedigitalcollection.asme.org/computingengineering/article-pdf/22/4/044501/6850327/jcise_22_4_044501.pdf. URL: <https://doi.org/10.1115/1.4053671>.
- [31] Tianyi Li et al. *Multi-scale Reconstruction of Turbulent Rotating Flows with Generative Diffusion Models*. 2023. arXiv: 2312.11121 [physics.flu-dyn]. URL: <https://arxiv.org/abs/2312.11121>.
- [32] Ning Liu and Guangtao Zhai. “Free Energy Adjusted Peak Signal to Noise Ratio (FEA-PSNR) for Image Quality Assessment”. In: *Sensing and Imaging* 18.1, 11 (Dec. 2017), p. 11. DOI: 10.1007/s11220-017-0160-0.
- [33] Vamsi Sai Krishna Malineni, Suresh Rajendran, and Kaushik Mitra. “A surrogate model based on Physics Informed Neural Networks for predicting wake flow past a cylinder”. In: *OCEANS 2023 - Limerick*. June 2023, pp. 1–10. DOI: 10.1109/OCEANS2023Limerick52467.2023.10244341.
- [34] Florian Menter et al. “An Overview of Hybrid RANS–LES Models Developed for Industrial CFD”. In: *Applied Sciences* 11.6 (2021). ISSN: 2076-3417. DOI: 10.3390/app11062459. URL: <https://www.mdpi.com/2076-3417/11/6/2459>.
- [35] Umberto Michelucci. *An Introduction to Autoencoders*. 2022. arXiv: 2201.03898 [cs.LG]. URL: <https://arxiv.org/abs/2201.03898>.

- [36] Ibomoiye Domor Mienye, Theo G. Swart, and George Obaido. “Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications”. In: *Information* 15.9 (2024). ISSN: 2078-2489. DOI: [10.3390/info15090517](https://doi.org/10.3390/info15090517). URL: <https://www.mdpi.com/2078-2489/15/9/517>.
- [37] Edmondo Minisci. *SURROGATE MODELLING TECHNIQUES*. 2024. URL: <https://www.sto.nato.int/publications/STO%20Educational%20Notes/STO-EN-AVT-385/EN-AVT-385-05.pdf>.
- [38] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. “Making a “Completely Blind” Image Quality Analyzer”. In: *IEEE Signal Processing Letters* 20.3 (2013), pp. 209–212. DOI: [10.1109/LSP.2012.2227726](https://doi.org/10.1109/LSP.2012.2227726).
- [39] F. Mohamadi and Ali Fazeli. “A Review on Applications of CFD Modeling in COVID-19 Pandemic”. In: *Archives of Computational Methods in Engineering* 29 (Jan. 2022). DOI: [10.1007/s11831-021-09706-3](https://doi.org/10.1007/s11831-021-09706-3).
- [40] Nina Morozova et al. “A CFD-based surrogate model for predicting flow parameters in a ventilated room using sensor readings”. In: *Energy and Buildings* 266 (2022), p. 112146. ISSN: 0378-7788. DOI: <https://doi.org/10.1016/j.enbuild.2022.112146>. URL: <https://www.sciencedirect.com/science/article/pii/S0378778822003176>.
- [41] M.C. Neves et al. “Enhancing experimental image quality in two-phase bubbly systems with super-resolution using generative adversarial networks”. In: *International Journal of Multiphase Flow* 180 (2024), p. 104952. ISSN: 0301-9322. DOI: <https://doi.org/10.1016/j.ijmultiphaseflow.2024.104952>. URL: <https://www.sciencedirect.com/science/article/pii/S0301932224002295>.
- [42] Venant Niyonkuru. “Computational Fluid Dynamics (CFD) for blood flow in cardiovascular medical devices and blood damage prediction”. In: *Journal of Preventive and Complementary Medicine* 2.3 (2023), pp. 136–142. ISSN: 3041-9700. DOI: [10.22034/ncm.2023.408048.1101](https://doi.org/10.22034/ncm.2023.408048.1101). eprint: https://www.jpccmed.com/article_177990_elab1d3b43b80ec00b6be1084f1feae7.pdf. URL: https://www.jpccmed.com/article_177990.html.
- [43] Keiron O’Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: [1511.08458](https://arxiv.org/abs/1511.08458) [cs.NE]. URL: <https://arxiv.org/abs/1511.08458>.
- [44] Francis Ogoke et al. *Inexpensive High Fidelity Melt Pool Models in Additive Manufacturing Using Generative Deep Diffusion*. 2023. arXiv: [2311.16168](https://arxiv.org/abs/2311.16168) [cs.LG]. URL: <https://arxiv.org/abs/2311.16168>.
- [45] Yidong Ouyang et al. *Transfer Learning for Diffusion Models*. 2024. arXiv: [2405.16876](https://arxiv.org/abs/2405.16876) [cs.LG]. URL: <https://arxiv.org/abs/2405.16876>.
- [46] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. arXiv: [1912.01703](https://arxiv.org/abs/1912.01703) [cs.LG]. URL: <https://arxiv.org/abs/1912.01703>.
- [47] Ethan Perez et al. “FiLM: Visual Reasoning with a General Conditioning Layer”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (June 2018). DOI: [10.1609/aaai.v32i1.11671](https://doi.org/10.1609/aaai.v32i1.11671). URL: <https://ojs.aaai.org/index.php/AAAI/article/view/11671>.
- [48] Jiaheng Qi and Hongbing Ma. “A Combined Model of Diffusion Model and Enhanced Residual Network for Super-Resolution Reconstruction of Turbulent Flows”. In: *Mathematics* 12 (7 Apr. 2024). ISSN: 22277390. DOI: [10.3390/math12071028](https://doi.org/10.3390/math12071028).

- [49] César Quilodrán-Casas and Rossella Arcucci. “A data-driven adversarial machine learning for 3D surrogates of unstructured computational fluid dynamic simulations”. In: *Physica A: Statistical Mechanics and its Applications* 615 (Apr. 2023). ISSN: 03784371. DOI: [10.1016/j.physa.2023.128564](https://doi.org/10.1016/j.physa.2023.128564).
- [50] Pu Ren et al. *Physics-informed Deep Super-resolution for Spatiotemporal Data*. 2022. arXiv: [2208.01462](https://arxiv.org/abs/2208.01462) [cs.LG]. URL: <https://arxiv.org/abs/2208.01462>.
- [51] Pu Ren et al. *SuperBench: A Super-Resolution Benchmark Dataset for Scientific Machine Learning*. 2023. arXiv: [2306.14070](https://arxiv.org/abs/2306.14070) [cs.CV]. URL: <https://arxiv.org/abs/2306.14070>.
- [52] Ângela M. Ribau et al. “Flow Structures Identification through Proper Orthogonal Decomposition: The Flow around Two Distinct Cylinders”. In: *Fluids* 6.11 (2021). ISSN: 2311-5521. DOI: [10.3390/fluids6110384](https://doi.org/10.3390/fluids6110384). URL: <https://www.mdpi.com/2311-5521/6/11/384>.
- [53] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: [2112.10752](https://arxiv.org/abs/2112.10752) [cs.CV]. URL: <https://arxiv.org/abs/2112.10752>.
- [54] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: [1505.04597](https://arxiv.org/abs/1505.04597) [cs.CV]. URL: <https://arxiv.org/abs/1505.04597>.
- [55] Xavier Roynard, Fabien Casenave, and Brian Staber. *AirfRANS_original*. Zenodo, June 2024. DOI: [10.5281/zenodo.12515084](https://doi.org/10.5281/zenodo.12515084). URL: <https://doi.org/10.5281/zenodo.12515084>.
- [56] Chitwan Saharia et al. *Palette: Image-to-Image Diffusion Models*. 2022. arXiv: [2111.05826](https://arxiv.org/abs/2111.05826) [cs.CV]. URL: <https://arxiv.org/abs/2111.05826>.
- [57] Mohammed Sardar et al. *Spectrally Decomposed Diffusion Models for Generative Turbulence Recovery*. 2023. arXiv: [2312.15029](https://arxiv.org/abs/2312.15029) [physics.flu-dyn]. URL: <https://arxiv.org/abs/2312.15029>.
- [58] Shuyao Shang et al. “ResDiff: Combining CNN and Diffusion Model for Image Super-resolution”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.8 (Mar. 2024), pp. 8975–8983. DOI: [10.1609/aaai.v38i8.28746](https://doi.org/10.1609/aaai.v38i8.28746). URL: <https://ojs.aaai.org/index.php/AAAI/article/view/28746>.
- [59] Dule Shu, Zijie Li, and Amir Barati Farimani. “A physics-informed diffusion model for high-fidelity flow field reconstruction”. In: *Journal of Computational Physics* 478 (2023), p. 111972. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2023.111972>. URL: <https://www.sciencedirect.com/science/article/pii/S0021999123000670>.
- [60] Filippas Sofos and Dimitris Drikakis. “A review of deep learning for super-resolution in fluid flows”. In: *Physics of Fluids* 37.4 (Apr. 2025), p. 041303. DOI: [10.1063/5.0265738](https://doi.org/10.1063/5.0265738). eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0265738/20487460/041303_1_5.0265738.pdf. URL: <https://doi.org/10.1063/5.0265738>.
- [61] Ruan Sousa et al. *Data-driven diffusion-based super-resolution applied to Rayleigh-Bénard convection*. Nov. 2024.
- [62] Bruno Sudret, Stefano Marelli, and Joe Wiart. “Surrogate models for uncertainty quantification: An overview”. In: *2017 11th European Conference on Antennas and Propagation (EUCAP)*. 2017, pp. 793–797. DOI: [10.23919/EuCAP.2017.7928679](https://doi.org/10.23919/EuCAP.2017.7928679).

- [63] C. Bane Sullivan and Alexander A. Kaszynski. “PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)”. In: *Journal of Open Source Software* 4.37 (2019), p. 1450. DOI: [10.21105/joss.01450](https://doi.org/10.21105/joss.01450). URL: <https://doi.org/10.21105/joss.01450>.
- [64] Sepehrdad Tahmasebi et al. *Using Diffusion Models for Reducing Spatiotemporal Errors of Deep Learning Based Urban Microclimate Predictions at Post-Processing Stage*. 2025. arXiv: 2501.04847 [physics.flu-dyn]. URL: <https://arxiv.org/abs/2501.04847>.
- [65] Y. Toparlar et al. “A review on the CFD analysis of urban microclimate”. In: *Renewable and Sustainable Energy Reviews* 80 (2017), pp. 1613–1640. ISSN: 1364-0321. DOI: <https://doi.org/10.1016/j.rser.2017.05.248>. URL: <https://www.sciencedirect.com/science/article/pii/S1364032117308924>.
- [66] Ricardo Vinuesa and Steven L. Brunton. “Enhancing computational fluid dynamics with machine learning”. In: *Nature Computational Science* 2.6 (June 2022), pp. 358–366. ISSN: 2662-8457. DOI: [10.1038/s43588-022-00264-7](https://doi.org/10.1038/s43588-022-00264-7). URL: <http://dx.doi.org/10.1038/s43588-022-00264-7>.
- [67] Ricardo Vinuesa and Steven L. Brunton. “The Potential of Machine Learning to Enhance Computational Fluid Dynamics”. In: *ArXiv abs/2110.02085* (2021). URL: <https://api.semanticscholar.org/CorpusID:238354390>.
- [68] Rui Wang et al. *Towards Physics-informed Deep Learning for Turbulent Flow Prediction*. 2020. arXiv: 1911.08655 [physics.comp-ph]. URL: <https://arxiv.org/abs/1911.08655>.
- [69] Xiyu Wang et al. *Efficient Transfer Learning in Diffusion Models via Adversarial Noise*. 2023. arXiv: 2308.11948 [cs.CV]. URL: <https://arxiv.org/abs/2308.11948>.
- [70] Maximilian Werhahn et al. “A Multi-Pass GAN for Fluid Flow Super-Resolution”. In: *Proc. ACM Comput. Graph. Interact. Tech.* 2.2 (July 2019). DOI: [10.1145/3340251](https://doi.org/10.1145/3340251). URL: <https://doi.org/10.1145/3340251>.
- [71] Brett Wilson, Sarah Wakes, and Michael Mayo. “Surrogate modeling a computational fluid dynamics-based wind turbine wake simulation using machine learning”. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2017, pp. 1–8. DOI: [10.1109/SSCI.2017.8280844](https://doi.org/10.1109/SSCI.2017.8280844).
- [72] Haizhou WU et al. “A generative deep learning framework for airfoil flow field prediction with sparse data”. In: *Chinese Journal of Aeronautics* 35.1 (2022), pp. 470–484. ISSN: 1000-9361. DOI: <https://doi.org/10.1016/j.cja.2021.02.012>. URL: <https://www.sciencedirect.com/science/article/pii/S1000936121000728>.
- [73] Hanguang Xiao et al. “Deep learning for medical imaging super-resolution: A comprehensive review”. In: *Neurocomputing* 630 (2025), p. 129667. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2025.129667>. URL: <https://www.sciencedirect.com/science/article/pii/S092523122500339X>.
- [74] Wenzhuo Xu, Noelia Grande Gutierrez, and Christopher McComb. “MegaFlow2D: A Parametric Dataset for Machine Learning Super-resolution in Computational Fluid Dynamics Simulations”. In: *Proceedings of Cyber-Physical Systems and Internet of Things Week 2023. CPS-IoT Week’ 23*. San Antonio, TX, USA: Association for Computing Machinery, 2023, pp. 100–104. ISBN: 9798400700491. DOI: [10.1145/3576914.3587552](https://doi.org/10.1145/3576914.3587552). URL: <https://doi.org/10.1145/3576914.3587552>.

- [75] Xiaosen Xu et al. “A case study of offshore wind turbine positioning optimization methodology using a novel multi-stage approach”. In: *Frontiers in Marine Science* Volume 9 - 2022 (2022). ISSN: 2296-7745. DOI: [10.3389/fmars.2022.1028732](https://doi.org/10.3389/fmars.2022.1028732). URL: <https://www.frontiersin.org/journals/marine-science/articles/10.3389/fmars.2022.1028732>.
- [76] Gefan Yang and Stefan Sommer. *A Denoising Diffusion Model for Fluid Field Prediction*. 2023. arXiv: [2301.11661](https://arxiv.org/abs/2301.11661) [cs.LG]. URL: <https://arxiv.org/abs/2301.11661>.
- [77] Hongqian Yu et al. “A transfer learning method to assimilate numerical data with experimental data for effusion cooling”. In: *Applied Thermal Engineering* 224 (2023), p. 120075. ISSN: 1359-4311. DOI: <https://doi.org/10.1016/j.applthermaleng.2023.120075>. URL: <https://www.sciencedirect.com/science/article/pii/S1359431123001047>.
- [78] Linqi Yu et al. “Three-dimensional ESRGAN for super-resolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning”. In: *Physics of Fluids* 34.12 (Dec. 2022), p. 125126. ISSN: 1070-6631. DOI: [10.1063/5.0129203](https://doi.org/10.1063/5.0129203). eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0129203/16611903/125126_1_online.pdf. URL: <https://doi.org/10.1063/5.0129203>.
- [79] Frank J. Zajackowski, Sue Ellen Haupt, and Kerrie J. Schmehl. “A preliminary study of assimilating numerical weather prediction data into computational fluid dynamics models for wind prediction”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 99.4 (2011). The Fifth International Symposium on Computational Wind Engineering, pp. 320–329. ISSN: 0167-6105. DOI: <https://doi.org/10.1016/j.jweia.2011.01.023>. URL: <https://www.sciencedirect.com/science/article/pii/S0167610511000250>.
- [80] M. H. Zawawi et al. “A review: Fundamentals of computational fluid dynamics (CFD)”. In: *AIP Conference Proceedings* 2030.1 (Nov. 2018), p. 020252. ISSN: 0094-243X. DOI: [10.1063/1.5066893](https://doi.org/10.1063/1.5066893). eprint: https://pubs.aip.org/aip/acp/article-pdf/doi/10.1063/1.5066893/7654457/020252_1_online.pdf. URL: <https://doi.org/10.1063/1.5066893>.
- [81] Jinsong Zhang, Jian Liu, and Zhenwei Huang. “Improved deep learning method for accurate flow field reconstruction from sparse data”. In: *Ocean Engineering* 280 (2023), p. 114902. ISSN: 0029-8018. DOI: <https://doi.org/10.1016/j.oceaneng.2023.114902>. URL: <https://www.sciencedirect.com/science/article/pii/S0029801823012866>.
- [82] Lin Zhang et al. “FSIM: A Feature Similarity Index for Image Quality Assessment”. In: *IEEE Transactions on Image Processing* 20.8 (2011), pp. 2378–2386. DOI: [10.1109/TIP.2011.2109730](https://doi.org/10.1109/TIP.2011.2109730).
- [83] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. arXiv: [1801.03924](https://arxiv.org/abs/1801.03924) [cs.CV]. URL: <https://arxiv.org/abs/1801.03924>.
- [84] Chenxu Zheng, Weiming Huang, and Wenjiang Xu. “Learning-Based Super-Resolution Imaging of Turbulent Flames in Both Time and 3D Space Using Double GAN Architectures”. In: *Fire* 7.8 (2024). ISSN: 2571-6255. DOI: [10.3390/fire7080293](https://doi.org/10.3390/fire7080293). URL: <https://www.mdpi.com/2571-6255/7/8/293>.