

ABSTRACT

Title of thesis: **DENOISING THE DESIGN SPACE:
DIFFUSION MODELS FOR ACCELERATED
AIRFOIL SHAPE OPTIMIZATION**

Cashen Diniz, Master of Science, 2024

Thesis directed by: **Professor Mark D. Fuge**
Department of Mechanical Engineering

Generative models offer the possibility to accelerate and potentially substitute parts of the often expensive traditional design optimization process. We present Aero-DDM, a novel application of a latent denoising diffusion model (DDM) capable of generating airfoil geometries conditioned on flow parameters and an area constraint. Additionally, we create a novel, diverse dataset ¹ of optimized airfoil designs that better reflects a realistic design space than has been done in previous work. Aero-DDM is applied to this dataset, and key metrics are assessed both statistically and with an open-source computational fluid dynamics (CFD) solver to determine the performance of the generated designs. We compare our approach to an optimal transport GAN, and demonstrate that our model can generate designs with superior performance statistically, in aerodynamic benchmarks, and in warm-start scenarios. We also extend our diffusion model approach, and demonstrate that the number of steps required for inference can be reduced by as much as $\sim 86\%$, compared to an optimized version of the baseline inference process, without meaningful degradation in design quality, simply by using the initial design to start the sampling process.

¹<https://github.com/IDEALLab/OptimizingDiffusionSciTech2024>

DENOISING THE DESIGN SPACE: DIFFUSION MODELS FOR
ACCELERATED AIRFOIL SHAPE OPTIMIZATION

by

Cashen Diniz

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2024

Advisory Committee:
Professor Mark D. Fuge, Chair/Advisor
Professor Michael Otte
Professor James Baeder

© Copyright by
Cashen Diniz
2024

Foreword

Portions of this thesis are taken from and based on the author's own previous work, "Optimizing Diffusion to Diffuse Optimal Design", originally published as a conference paper by the American Institute of Aeronautics and Astronautics [6].



2181 Glenn L. Martin Hall
College Park, MD 20742-3035
TEL: 301-405-2410
<http://www.enme.umd.edu>

4/18/2024

Re: Previously Published Materials appearing in Thesis or Dissertation

Dean of the Graduate School,

Cashen Diniz has:

☐ NOT INCLUDED any previously published works within their thesis or dissertation.

☒ INCLUDED one or more previously published works within their thesis or dissertation. This letter certifies that the examining committee for the student has determined that the student made a substantial contribution to the previously published work. The inclusion of the previously published work has the approval of the thesis or dissertation advisor and the Graduate Director.

Sincerely,

A handwritten signature in black ink, appearing to read "P. Sandborn".

Peter Sandborn
Director of Graduate Studies
Department of Mechanical Engineering
University of Maryland

Sincerely,

Mark Fuge

Mark Fuge
Advisor for Cashen Diniz

Digitally signed by Mark
Fuge
Date: 2024.04.18 12:08:51
-04'00'

Dedication

I dedicate this thesis to my dog, Coco, who has been at my side throughout the process of writing this thesis. Thanks for listening to all my crazy ideas and reminding me to have some fun.

Acknowledgments

I am extremely grateful to all those who have helped me get to the point of finishing this thesis. First and foremost, I would like to thank Dr. Mark Fuge, who has been an invaluable source of support, not only as an advisor, but as a colleague and a friend. Without his belief in me, I simply would not have been able to come this far. I would also like to thank Milad, Nathan, Nicholas, Arthur, Quiyi, and Xuiliang at the IDEAL lab for welcoming me, and for all the discussions we've had. I also want to thank my best friend Sean for keeping me motivated through and through. Finally, I want to thank my parents, my brother Odin, and everyone else in my family for being there for me at every step along the way.

Table of Contents

Foreword	ii
Dedication	iv
Acknowledgements	v
1 Introduction	1
2 Related Work	4
2.1 Approaches to Aerodynamic Shape Optimization	4
2.2 CEBGAN	5
2.2.1 Bezier Representation	6
2.3 Diffusion Model Theory	6
2.4 Key Metrics	8
2.4.1 Maximum Mean Discrepancy	9
2.4.2 Area Constraint Mean Squared Error (Area MSE)	9
2.4.3 Instantaneous Optimality Gap (IOG)	10
2.4.4 Cumulative Optimality Gap (COG)	10
2.4.5 Reduction in Cumulative Optimality Gap (RiCOG)	10
3 Optimization Problem Setup	12
3.1 Design Space	12
3.2 Problem Formulation	13
3.3 Solver Setup	15
3.4 Geometry Parameterization and Meshing	15
4 Dataset Results	18
4.1 Optimization Results	19
4.2 Data Analysis	22

5	Generative Model Methodology	27
5.1	CEBGAN Modifications	27
5.2	Aero-DDM Architecture	27
5.3	Bezier Autoencoder Validation	29
5.4	Diffusion Parameters and Scheduling	31
5.5	Benchmarking and Warm Start Optimization	32
5.6	DDM Inference Efficiency Study	33
6	Model Results	35
6.1	Sample Generations	35
6.2	Statistical Validation	37
6.3	DDM Inference Ablation Study	38
6.4	Aerodynamic Performance Benchmarks	41
6.5	Warm Start Performance	42
7	Conclusions	46
7.1	Future Work	47
A	Other Geometric Parameterization Methods	50
B	Example Optimization Trajectory Plots for a Single Case	51
C	Effect of T_{start} on Designs	53
	Bibliography	54

Chapter 1: Introduction

Airfoil design is an approachable, yet sufficiently difficult and realistic problem. Traditionally, airfoil design problems have been tackled by applying adjoint-based optimization methods, which can be implemented relatively efficiently and robustly [14]. Despite this, these solvers can be difficult for a non-expert to use properly and importantly, still require much in the way of computational resources for any large-scale design space study. In addition such gradient based solvers can easily converge to local minima, as the objective function is non-convex. To address some of these concerns, some authors have presented data-driven approaches to this problem [20, 39].

More recently, generative machine learning models have garnered significant attention in both academic and commercial circles. Generative models, when applied to engineering problems allow for the modeling of diverse, multi-modal distributions, and depending on the model used, can do so in a relatively data-efficient manner. One particular type of generative model, the denoising diffusion model (DDM) has recently surpassed other models such as generative adversarial networks (GANs) in image generation tasks [5]. Notable examples of such models include OpenAI’s DALL-E2 and Stability AI’s Stable Diffusion [30, 33]. Generative models have also demonstrated their efficacy in various engineering optimization design problems, either as a replacement or complement to traditional approaches. This is because engineering problems often exhibit multimodal behavior, high dimensionality, and non-intuitive design spaces [29, 4]. However, applying generative models to engineering design problems presents unique challenges due to the

need to satisfy physical constraints and simulate designs, making the task more complex than simpler image processing or generation tasks. Regenwetter *et al.* [31] identified four key challenges to applying generative models in engineering design, namely data sparsity, modeling design performance, feasibility, and sample diversity or model creativity.

DDMs have shown promise in image generation and processing tasks, outperforming GANs with generally higher quality and more diverse outputs [5]. DDMs are also relatively more stable to train than GANs, as they do not rely on an adversarial/game theoretic approach. Although DDMs are relatively new to the field of engineering design, early research suggests advantages in topology optimization. For example, in one study, DDMs reduced feasibility error by 11-fold and improved performance by a factor of 8 [26].

Several relevant publications have also applied generative models to design engineering problems. Ghosh *et al.* presented a probabilistic framework for generating airfoil designs using an invertible neural network trained on data generated by a real-valued non-volume preserving (real NVP) model. In this way, they were able to replace a traditional gradient-based iterative optimization approach. The real NVP model applies a sequence of probabilistic transformations to a latent input [9]. However, most publications apply a GAN to the task airfoil shape optimization in favor of a probabilistic approach. For example, Du *et al.* applied a GAN to efficiently reconstruct the (unconditional) UIUC dataset of airfoil designs [7]. Achour *et al.* built on the airfoil-generating GAN architecture to add conditioning to generations, with an intent to replace aspects of the traditional optimization pipeline [1]. Separately, Morton *et al.* have applied conditional GANs to predict the flow around circles and half-cylinders [28]. Although this work was not focused on the design optimization process, it demonstrates the broad applicability of generative models to the more general issue of parameterized fluid flow problems.

Our work focuses on the problem of airfoil shape optimization, in particular, we follow the conditional generative model work of Chen *et al.* who used a Conditional GAN to

generate airfoil designs based on a given Mach number (Ma), Reynold’s number (Re) and desired coefficient of lift (C_L) [2].

In this work we present four main contributions:

1. A diverse, robust dataset of optimized airfoils. We note the following novel contributions compared to previous work:
 - (a) An extensive description and analysis of our data and the collection process, which includes the use of a more robust FFD-based framework which has been well validated in literature. This includes the use of a structured, curvilinear meshing algorithm, as opposed to the unstructured meshes used in past work [17, 24, 34].
 - (b) Higher design diversity in the dataset; previous work has made use of optimized datasets which had a much greater degree of uni modal behaviour [2].
2. Presentation of a novel Diffusion model application to aero shape optimization, Aero-DDM, which has been tailored to the problem, and can generate designs conditional on both flow parameters, an initial design, and an area constraint.
3. A comparison of our Aero-DDM model to a state of the art optimal transport GAN model. We study statistical metrics, aerodynamic performance, and warm start efficiency.
4. An inference efficiency study of our Aero-DDM model, which demonstrates a reduction in the number of sampling steps required by $\sim 86\%$ compared to an optimized version of the baseline inference process, without loss in performance. The method we propose is simple, yet offers immense benefits for rapid design space exploration, considering how traditionally expensive inference can be for diffusion models in general.

Chapter 2: Related Work

Although we have chosen to add the context of related works throughout portions of this thesis, in this chapter we will address key areas which help explain our motivation and some of the background behind our work.

2.1 Approaches to Aerodynamic Shape Optimization

Aerodynamic shape optimization is the process of tuning a set of geometric design variables, $\mathbf{x}$, typically associated with a lifting body surface, to achieve improvement in some objective function, J subject to constraints. Effectively, this is a constrained minimization problem, and as such, can be solved with a wide array of methods depending on the specific problem. As J is often a performance metric, such as lift or drag, a flow solution, obtained through a computational fluid dynamics solver (CFD) may be required to achieve any level of accuracy. The computational cost of the simulations can be expensive, and therefore rules out a brute force search of the design space to find an optimal design [21]. One of the most efficient methods developed to tackle this problem is the adjoint optimization method [18]. The main benefit the adjoint method, is that the sensitivity of J with respect to the design variables, $\mathbf{x}$ can be computed at a cost which scales independently of the number of design variables. The sensitivity information can then be used to inform the change in the current design variables according to a chosen optimization algorithm [16].

Although the adjoint approach may be appealing for a wide range of problems, the

computational cost of evaluations can still prohibit it from being used as a tool for rapid, interactive, exploration of designs early on in the engineering process. As a result, many machine learning and surrogate based methods have been developed to fill in the gaps. Surrogate methods, which attempt to approximate after being trained on example data, can provide evaluations of key metrics in fractions of the amount of time it would take to run a full simulation or adjoint optimization. If the surrogate model is differentiable, which is the case for many neural-network based approaches, it can be substituted for the CFD evaluation in a gradient based optimization problem in order to generate new designs [21, 36].

An alternative to surrogate approaches is the use of generative models for design. These models can generate high quality designs without the need to solve a surrogate optimization problem, as the input to the model can include conditional information about the flow regime or required constraints. Popular generative models include GANs, Variational Autoencoders, Flow-based models, and diffusion models [13]. As discussed in the introduction, of these approaches, GANs have seen the most use in the field of aerodynamic shape optimization, specifically with application to airfoil shape optimization.

2.2 CEBGAN

Much of the work presented in this thesis focuses on a comparison of our implementation to a GAN model developed Chen et. al. The model they developed, the conditional entropic Bezier GAN, CEBGAN, is a discriminator-free GAN which relies on optimal transport based loss to learn the airfoil distribution. We chose this comparison because CEBGAN was shown to outperform traditional GANs in airfoil generation, and in general produced state-of-the-art performance on the task. In addition, the authors of CEBGAN applied a Sinkhorn divergence algorithm to overcome the issue of mode collapse, which

is a common issue with traditional GANs where the generator produces outputs with very little variation. In similar diffusion model applications to design tasks, traditional GANs are still often used in comparisons, despite being outclassed by more recent GAN variants [2].

2.2.1 Bezier Representation

A central component of CEBGAN, and an attribute shared with our Aero-DDM model, is the use of a latent Bezier representation for the airfoil coordinates. This idea was originally developed by the same authors in [3]. The general Bezier formulation, describing the coordinates, Y in terms of control points, cp , and weights, w , is as follows;

$$Y_j = \frac{\sum_i^N \binom{n}{i} t_j^i (1-t_j)^{n-i} cp_i w_i}{\sum_i^N \binom{n}{i} t_j^i (1-t_j)^{n-i} w_i} \quad (2.1)$$

[3].

Where j varies between 0 and the number of points, m . $t \in [0, 1]$ is a parametric variable which discretizes the curve, and N is the degree of the Bezier polynomial. In their work, Chen. et. al found that Bezier representations offered smoother, more feasible designs in a lower dimensional space compared to a GAN trained on raw coordinates [3]. Finally, we direct the reader to the appendix A for a brief discussion of alternative geometric parameterizations, which may be useful for future work.

2.3 Diffusion Model Theory

Diffusion models are part of a greater family of models, all of which are based on the idea of maximizing the likelihood, $p(\mathbf{x})$, of all known data, $\mathbf{x}$. In practical problems, the ground-truth function describing $p(\mathbf{x})$ is often complex, and $\mathbf{x}$ can also be quite high-

dimensional. As such, learning $p(\mathbf{x})$ exactly can be computationally infeasible. Therefore, likelihood-maximizing models instead introduce a random latent variable, $\mathbf{z}$, of lower-dimensional, and or lower complexity, which can be used to describe the joint distribution with $\mathbf{x}$,

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}. \quad (2.2)$$

$$(2.3)$$

To realize the benefits of introducing the lower complexity $\mathbf{z}$, a tractable Evidence Lower Bound (ELBO) can be defined to approximate the joint distribution integral;

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \quad (2.4)$$

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \quad (2.5)$$

$$= \log \int \frac{p(\mathbf{x}, \mathbf{z}) q_\theta(\mathbf{z}|\mathbf{x})}{q_\theta(\mathbf{z}|\mathbf{x})} d\mathbf{z} \quad (2.6)$$

$$= \log \mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})} \left[\frac{p(\mathbf{x}, \mathbf{z})}{q_\theta(\mathbf{z}|\mathbf{x})} \right] \quad (2.7)$$

$$\geq \mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})} \left[\frac{p(\mathbf{x}, \mathbf{z})}{q_\theta(\mathbf{z}|\mathbf{x})} \right] \quad (\text{Using Jensen's Inequality}) \quad (2.8)$$

$$(2.9)$$

Adapted from Luo: [23, 6]

Where $q_\theta(\mathbf{z}|\mathbf{x})$ is the variational distribution with learnable model parameters θ [23].

Diffusion models differ from other related likelihood maximizing approaches in that $\mathbf{z}$, has the same cardinality as the data $\mathbf{x}$, but is noised according to a variance schedule

parameterized by a hyperparameter, β_t . The index t describes the data-to-noise ratio and ranges from 0 to T , with 0 representing the original, un-noised data, and T representing maximally noised data. In the limit as T increases, the data approaches an isotropic Gaussian distribution;

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, 1 - \alpha_t) \quad (2.10)$$

Here, $\alpha_t = 1 - \beta_t$.

The act of noising the original data according to a parameterized Gaussian is known as the *forward* process. The *reverse* process can be computed using the model predictions;

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad (2.11)$$

where,

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t)) \quad (2.12)$$

[23]. Commonly, σ_θ is set equal to β_t . The final ELBO objective can be written as

$$ELBO = \mathbb{E}_{q(\mathbf{x}_{0:T}|\mathbf{x}_0)} \left[\frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \quad (2.13)$$

In many cases, including this thesis, the MSE between the predicted and actual noise added to the data can be used as a much simpler estimation of the ELBO [15].

2.4 Key Metrics

Throughout this thesis, we will refer to various statistical, and performance measures and related terms which we will briefly discuss here. We specifically chose to use many

metrics which were also used by Chen et. al. in [2], as it creates a degree of consistency between both studies.

2.4.1 Maximum Mean Discrepancy

Maximum mean discrepancy (MMD) is a metric which attempts to measure the similarity between two distributions, and is commonly used to determine the performance of generative machine learning models. It is defined as

$$\text{MMD}^2(P, Q) = \mathbb{E}_{x, x' \sim P}[k(x, x')] + \mathbb{E}_{y, y' \sim Q}[k(y, y')] - 2\mathbb{E}_{x \sim P, y \sim Q}[k(x, y)] \quad (2.14)$$

. Where P and Q are the distributions we are interested in, x and y are independent random variables with distributions P , and Q respectively. x' and y' are additional independent copies of x and y with the same P and Q distributions. k is a chosen kernel function - in this thesis we use a Gaussian kernel with a σ of 1 [11].

2.4.2 Area Constraint Mean Squared Error (Area MSE)

In parts of this thesis we attempt to measure violation of the area constraint specific to our problem. Our formulation of this metric is simple; prescribing a value of zero unless the area of the generated airfoil is less than the minimum area constraint, in which case the squared error is applied:

$$A_{Error} = \begin{cases} 0 & \text{if } A_{gen} \geq A_{min} \\ ||A_{min} - A_{gen}||^2 & \text{if } A_{gen} < A_{min} \end{cases} \quad (2.15)$$

2.4.3 Instantaneous Optimality Gap (IOG)

Instantaneous optimality gap is a metric which attempts to measure the difference in performance between an initial design, x_0 , with objective function value J_0 , and the optimal design, x^* , with the objective function value J^* :

$$\text{IOG}(J) = \frac{|J_0 - J^*|}{J^*}. \quad (2.16)$$

While some formulations in literature choose to use a simple ratio, this can be misleading, as high objective ratios may be achieved by violating certain constraints. This is especially true for aerodynamic efficiency quantities such as $\frac{C_l}{C_d}$ where induced drag may spike even with a relatively minor change in C_l . Evidence for this can be found in the highly nonlinear drag polars from experiments and simulations in literature [32, 22].

2.4.4 Cumulative Optimality Gap (COG)

Cumulative optimality gap (COG) is a measure of how efficiently an optimizer is approaching an optimal design, x^* , at the current step i in an optimization trajectory, taking into consideration the values of all previous objective function evaluations:

$$\text{COG}(J) = \sum_i^N \frac{J^* - J_i}{J^*} \quad (2.17)$$

[2].

If $i = n$, COG measures the total efficiency of an optimization trajectory.

2.4.5 Reduction in Cumulative Optimality Gap (RiCOG)

Reduction in cumulative optimality gap (RiCOG) is similar to the COG metric, but instead measures optimizer efficiency *relative* to a baseline COG trajectory, J_o :

$$\text{RiCOG}(J) = \frac{\text{COG}(J_o) - \text{COG}(J)}{\text{COG}(J_o)} \times 100 \quad (2.18)$$

In this thesis, J_o is always the objective function optimization trajectory of our original adjoint optimized dataset.

Chapter 3: Optimization Problem Setup

Note: This chapter was taken from the author’s original work as published in [6].

We formulate a diverse, multi-modal, optimization problem by first specifying the flow conditions; Mach number (M) and Reynolds’s number (Re). In addition, we require the final design to attain the specified coefficient of lift (C_l) while keeping the area above the minimum area $\left(\left(\frac{A}{A_{init}}\right)_{min}\right)$ constraint. Thickness constraints are imposed based on the initial design—although we note that in practice, none of the adjoint solver iterations activated these constraints. We also impose no-twist leading and trailing edge conditions.

After defining the problem, we used Latin-hypercube sampling to generate the input parameter space, within the bounds specified in Table 3.1.

3.1 Design Space

Table 3.1: Sampled Parameter Bounds

Category	Parameter	Lower	Upper	Description
Flow Condition	M	0.4	0.9	Mach Number
	Re	1E6	10E6	Reynold’s Number
Constraint	C_l^{con}	0.5	1.2	Coefficient of Lift
	$\left(\frac{A}{A_{init}}\right)_{min}$	0.75	1.0	Minimum Area Fraction

The bounds were chosen purposefully to cover part of the transonic regime which is known to be difficult to model with non-RANS approaches, such as panel methods [20].

3.2 Problem Formulation

We define the specific optimization problem below in equation 3.1.

$$\begin{aligned}
& \min_{\Delta y_i, \alpha} C_d \\
& \text{s.t. } C_l = C_l^{con} \\
& \quad -0.025 \leq \Delta y_i \leq 0.025 \\
& \quad 0.0 \leq \alpha \leq 10.0 \\
& \quad \left(\frac{A}{A_{init}} \right)_{min} \leq \frac{A}{A_{init}} \leq 1.2 \\
& \quad 0.15 \leq \frac{t}{t_{init}} \leq 3.0 \\
& \quad t_{TE} = t_{TE,init} \\
& \quad y_{TE}^{upper} = -y_{TE}^{lower} \\
& \quad y_{LE}^{upper} = -y_{LE}^{lower}
\end{aligned} \tag{3.1}$$

Where Δy_i is the difference in the y-coordinates of the Free-Form Deformation (FFD) cage from the initial FFD cage. As such, Δy_i , has a large impact on how diverse the resulting design space is, and its value can be thought of as being embedded in the initial airfoil shape. Table 3.2 describes each of the optimization parameters in detail.

Table 3.2: Optimization Problem Parameters

Category	Parameter	Quantity	Lower	Upper	Units	Description
Objective	C_d	1	-	-	Non-Dim./Counts	Coefficient of Drag
Variable	Δy_i	20	-0.025	0.025	m	Change from initial FFD cage y value: $\Delta y_i = y_i - y_{init}$
	α	1	0.0	10.0	Degrees	Angle of Attack
Constraint	$C_l = C_l^{con}$	1	0.0	0.0	Non-Dim.	Coefficient of Lift
	$\frac{A}{A_{init}}$	1	$\left(\frac{A}{A_{init}}\right)_{min}$	1.20	Non-Dim.	Area Fraction; Relative to Initial
	$\frac{t}{t_{init}}$	100	0.15	3.00	Non-Dim.	Thickness Fraction; Relative to Initial
	$t_{TE} = t_{TE,init}$	1	0.0	0.0	m	Trailing Edge Thickness: Blunted TE Condition
	$y_{TE}^{upper} = -y_{TE}^{lower}$	1	0.0	0.0	m	Trailing Edge FFD Point Shearing Twist Condition
	$y_{LE}^{upper} = -y_{LE}^{lower}$	1	0.0	0.0	m	Leading Edge FFD Point Shearing Twist Condition

In addition to the scalar optimization parameters, 1400 initial airfoil designs were taken from a random subset of the UIUC dataset [27]. Raw coordinates from the dataset were re-sampled, smoothed, and normalized as part of a pre-processing procedure. Samples from this dataset are shown in Figure 3.1.

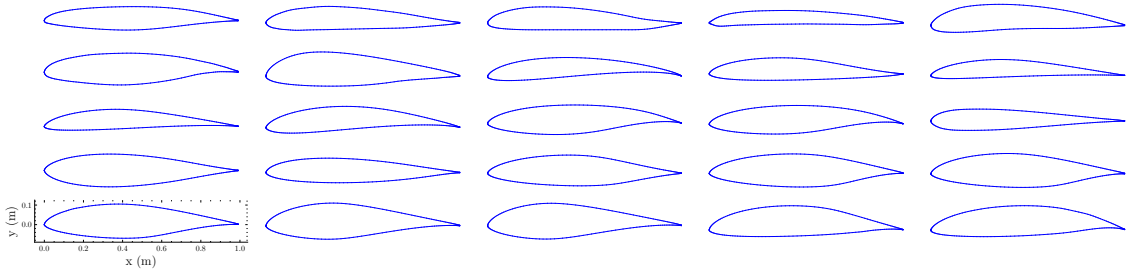


Figure 3.1: Processed Samples From UIUC Dataset

3.3 Solver Setup

We used the MACH-Aero framework for all discussed simulations¹. Within the MACH-Aero framework, structured mesh generation was handled automatically by pyHyp, a hyperbolic mesh generator [34]. Additional details concerning the mesh setup are described in the next section. pyOptSparse, a sparse numerical optimization suite, was used to efficiently solve the nonlinear optimization problem [37]. Specifically, we use the sequential least squares programming algorithm (SLSQP). pyGeo, a geometry parameterization module, was used to parameterize and deform the FFD cages, with warping of the cages handled by the robust IDWarp framework [12, 35]. The flow simulation, and the corresponding adjoint solution, was computed using the open-source ADflow solver [24, 17]. Within the solver, we used the approximate Newton–Krylov (ANK) method for improved convergence and robustness [38]. ADflow was configured to run Reynolds Averaged Navier Stokes (RANS) simulations with the Spalart-Allmaras model for turbulence effects.

For the flow solver we required that the L_2 norm of the residual decrease to $1e-8$, relative to the initial freestream flow residual. We use the same value for the adjoint residual convergence factor. For the optimization problem, we limit the maximum number of iterations to 500, and set a maximum step size of $1e-6$ for each of the design variables.

3.4 Geometry Parameterization and Meshing

Since we had to run a diverse set of input flow conditions and also initial airfoils, we decided to use a meshing setup that was able to successfully match the results from the benchmark ADODG RAE2822 airfoil optimization problem. A more detailed description of this benchmark problem can be found in [14]. The mesh used to successfully replicate

¹<https://github.com/mdolab/MACH-Aero>

the results from this benchmark problem, as well as the corresponding FFD cage is shown in Figure 3.2.

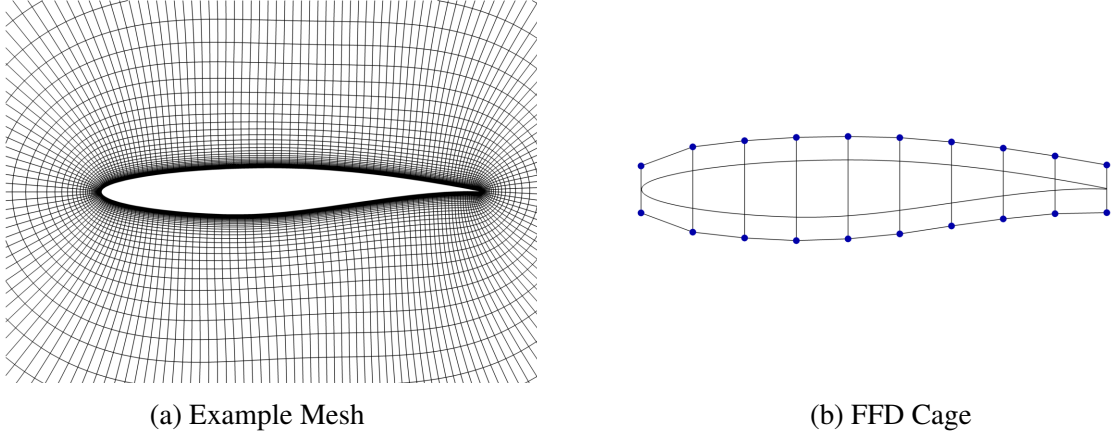
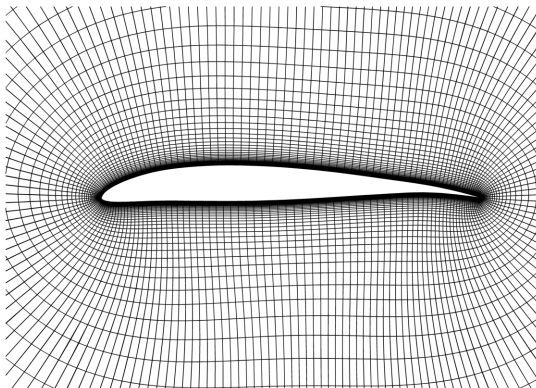
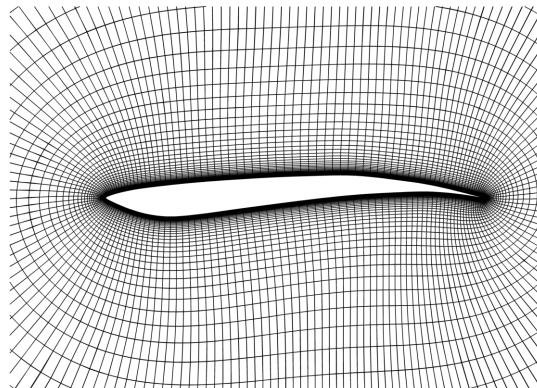


Figure 3.2: Example Setup: RAE2822

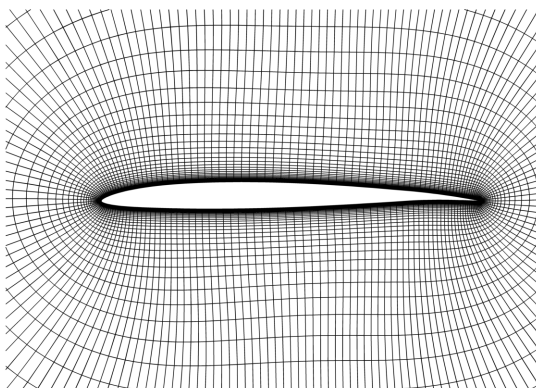
We set the far-field distance to 100 chord lengths away from each airfoil, and extrude the mesh using 100 grid levels. For each optimization run, we set the number of FFD points to 20 based on the results in [14], which showed reduced benefit after about 20 points were used. Finally, to improve stability we blunt the trailing edge at the point 99% away from the leading edge. Figure 3.3 shows the meshes for four random samples in our dataset after applying the described setup.



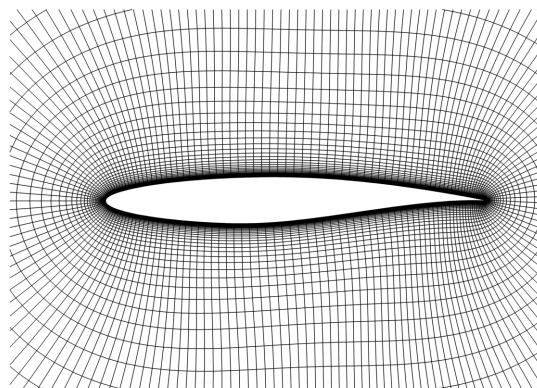
(a) Sample a



(b) Sample b



(c) Sample c



(d) Sample d

Figure 3.3: Four Random Samples: Meshing Setup

Chapter 4: Dataset Results

Note: This chapter was taken from the author’s original work as published in [\[6\]](#).

From the starting dataset of 1400 airfoils, we observed 935 successful optimizations, corresponding to a 67% success rate. In general, there were three main ways cases could fail. Firstly, they could fail due to issues with the flow solver. This could include meshing failures, such as the consistent presence of negative mesh volumes or highly irregular airfoil designs that were the result of large FFD cage changes. In these cases the flow solver could fail outright, or the bad flow solutions could make the Hessian ill-conditioned and cause the optimizer not to converge. The second failure mode was observed in the optimization process, with the most common error being a positive directional derivative in the linesearch. Although there is probably not one single reason for this, this could indicate that the problem has constraints which make a sufficient reduction in the objective infeasible. Finally, certain cases failed due to technical issues on the HPC system - although most of these cases were automatically restarted, and therefore this failure mode made up only a small portion of all failed cases.

The training, testing, and validation sets were split into 748, 140, and 47 airfoils respectively. We display a random subset of the resulting optimized airfoil dataset in [Figure 4.1](#).

Each optimization case was ran using 16 cores on a AMD EPYC 7763 processor, and took anywhere from 10-30 minutes to complete, depending on how many iterations were required for convergence, and on the specific performance of the specific CPU node at the

time. Batches of cases were submitted to be ran in parallel on an HPC system ¹. The entire set of simulations could take several days, and a few thousand service hours ($\sim 4k$), to complete. This includes the time required to submit and initialize jobs (in their virtual environments) and to post-process the results. We acknowledge some of the scaling issues could be greatly improved upon in the future.

4.1 Optimization Results

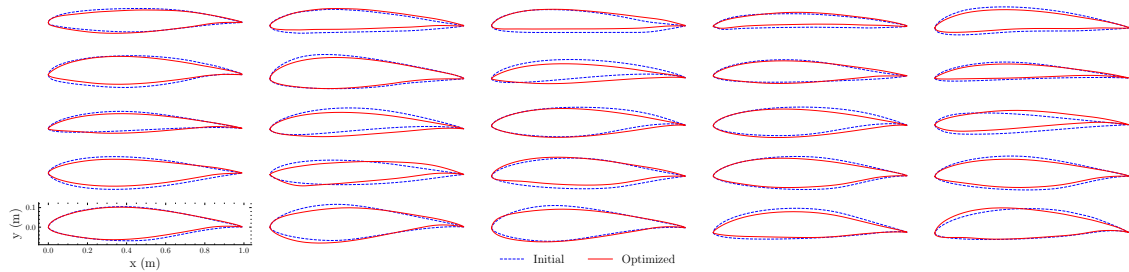


Figure 4.1: Examples of Initial and Optimized Airfoils

Of note is the relatively slight change in shape from the initial design. This can be explained by the proportionately small deviation constraint (± 0.025 m) from the initial FFD points we enforced on the optimizer. In testing, we found that making the range much larger made it difficult for the adjoint optimizer to converge consistently across the relatively wide range of flow parameters we sampled from. Techniques such as FFD point adaption, as described in [14], may be required for more dramatic changes to the design space.

Figure 4.2 presents more detailed results for the four aforementioned random samples in our dataset. In addition, Figure 4.3 shows the resulting pressure contours for the final, optimized designs. We note these samples display design characteristics one might expect for the given design parameters we used. For example, Figure 4.2 (b), corresponds to an

¹Zaratan Cluster; <https://hpcc.umd.edu/hpcc/zaratan.html>

optimized design with a relatively sharp, “diamond-shaped”, leading-edge profile typical of high Mach number airfoil designs, as such designs can better take advantage of the shock-expansion patterns relevant to transonic regimes [8].

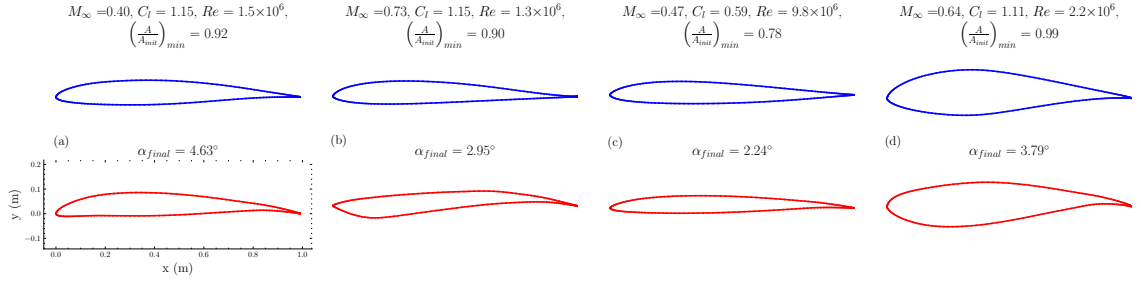


Figure 4.2: Four Random Samples: Optimized Airfoils

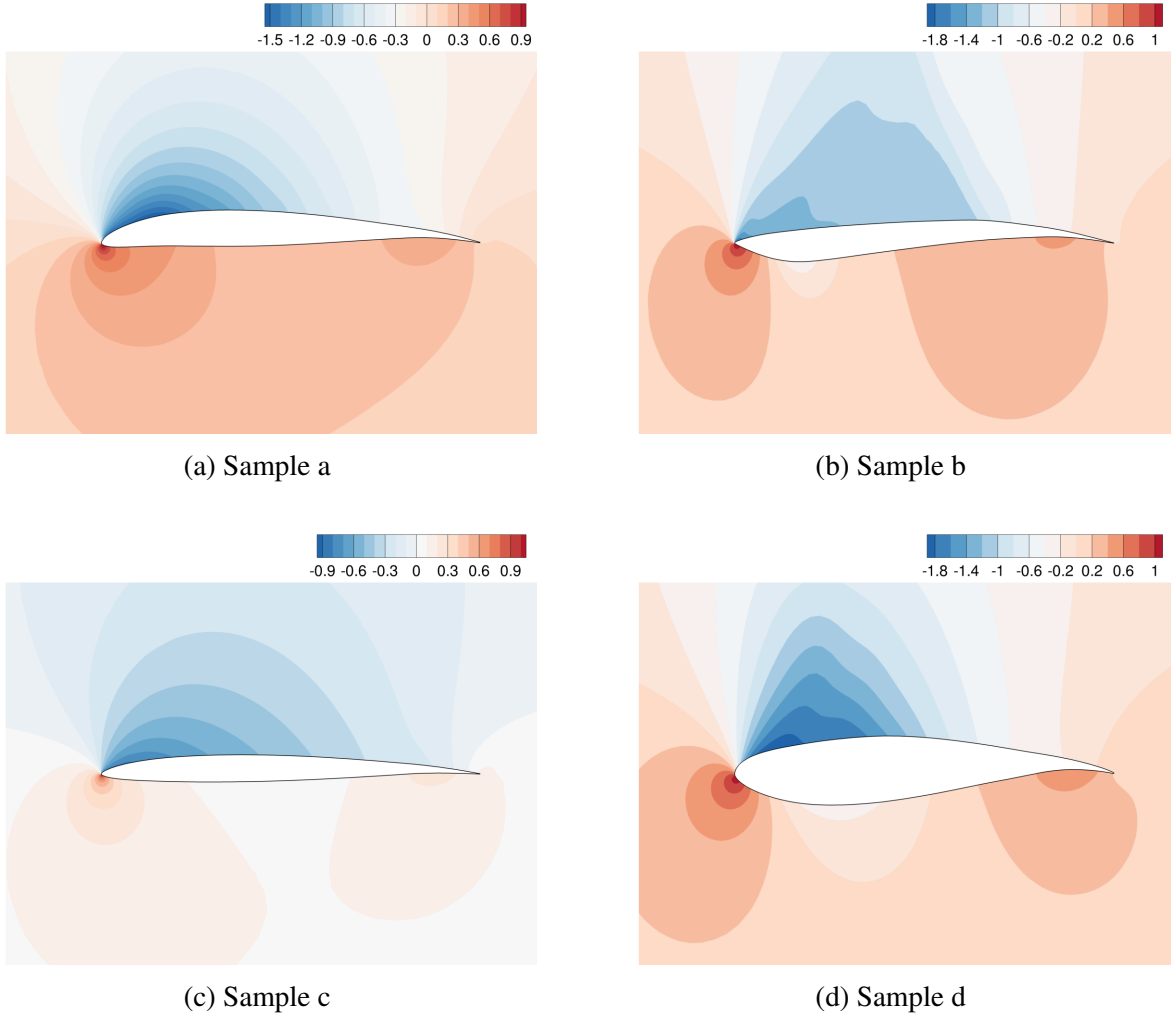


Figure 4.3: Four Random Samples: C_p Contours

Darker shades correspond to higher pressure differences relative to free stream conditions. Red and blue shades indicate higher and lower relative pressures respectively.

Finally, we present the optimization trajectory for the four random designs in Figure 4.4.

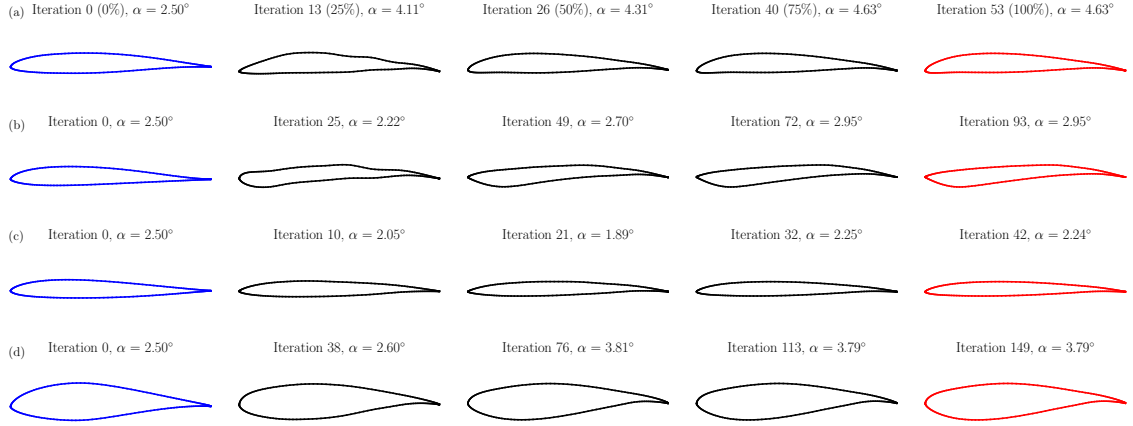


Figure 4.4: Four Random Samples: Optimization Trajectories

We make note of the fact that, beyond 25% of the optimization trajectory, the design stays relatively the same, with the latter steps only introducing marginal changes with a correspondingly smaller effect on C_d .

4.2 Data Analysis

To interpret all of the variable trajectory-length optimization data, we linearly interpolated the trajectory data to 1000 samples for all key objective, constraint, and design variables. Figure 4.5 shows the aggregate mean and standard deviation of the difference from the optimal objective, C_d^* , in drag counts (1 drag count equals 0.0001). As was the case visually in Fig. 4.4, Fig. 4.5 confirms that the change in optimal C_d is most affected by changes occurring in the first 20-25% of the design trajectory. We note that all the figures in this section display aggregate quantities computed across all 748 samples in the training set. In this section, σ will always refer to one standard deviation for a normal distribution centered at the mean value at each optimization iteration. Additionally, we direct the reader to appendix B for an example of an individual optimization trajectory for more context.

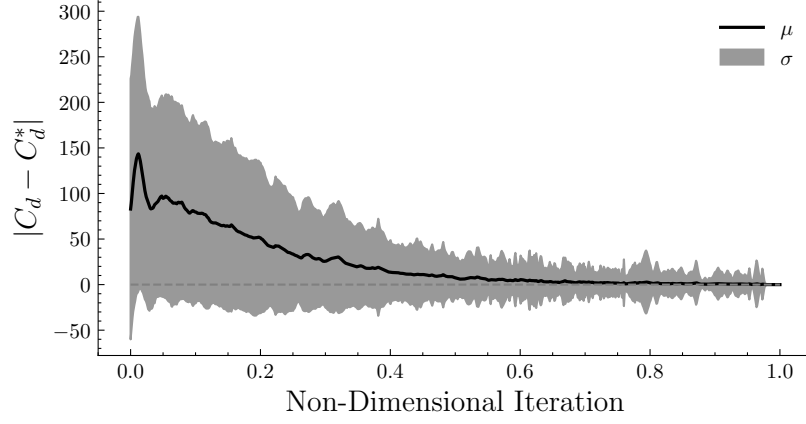


Figure 4.5: Aggregate C_d Convergence

The design variable optimization trajectories present a more or less linear decrease in the average per-iteration change for the majority of the trajectory, as can be seen for both angle of attack and in the summation of over the FFD y-coordinates in Figure 4.6. Where $\Delta Y_{n+1} - \Delta Y_n = \frac{1}{N_{ffd}} \sum_i^{N_{ffd}} |\Delta y_{n+1}^i - \Delta y_n^i|$. Note that this is the absolute change from the initial FFD cage, and is not the same as the change in coordinate value, which would produce a mean around zero due to enforcing symmetry in the FFD point movement (see 4.8). In this sense, a linear rate of change is reasonable, as we are not measuring any form of cumulative statistic.

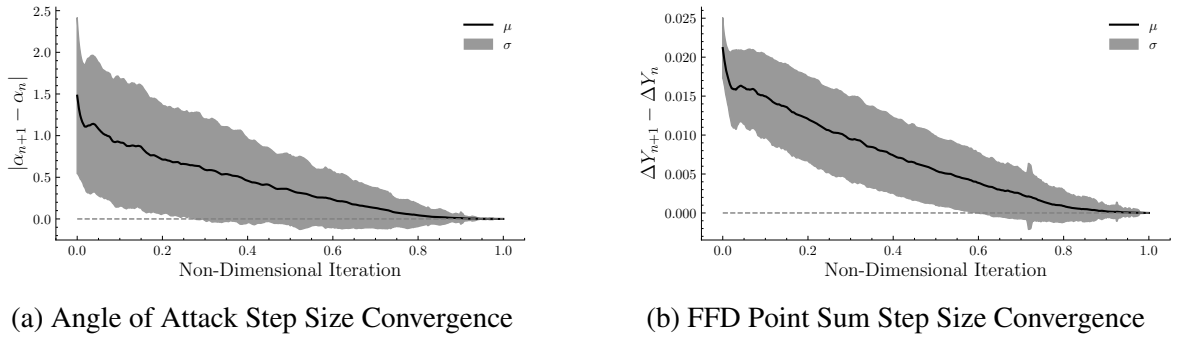


Figure 4.6: Aggregate Design Variable Step Size Convergence

Next, we present the aggregate constraint satisfaction measures for C_l , minimum area fraction, thickness constraints, and blunted thickness constraints in Figure 4.7.

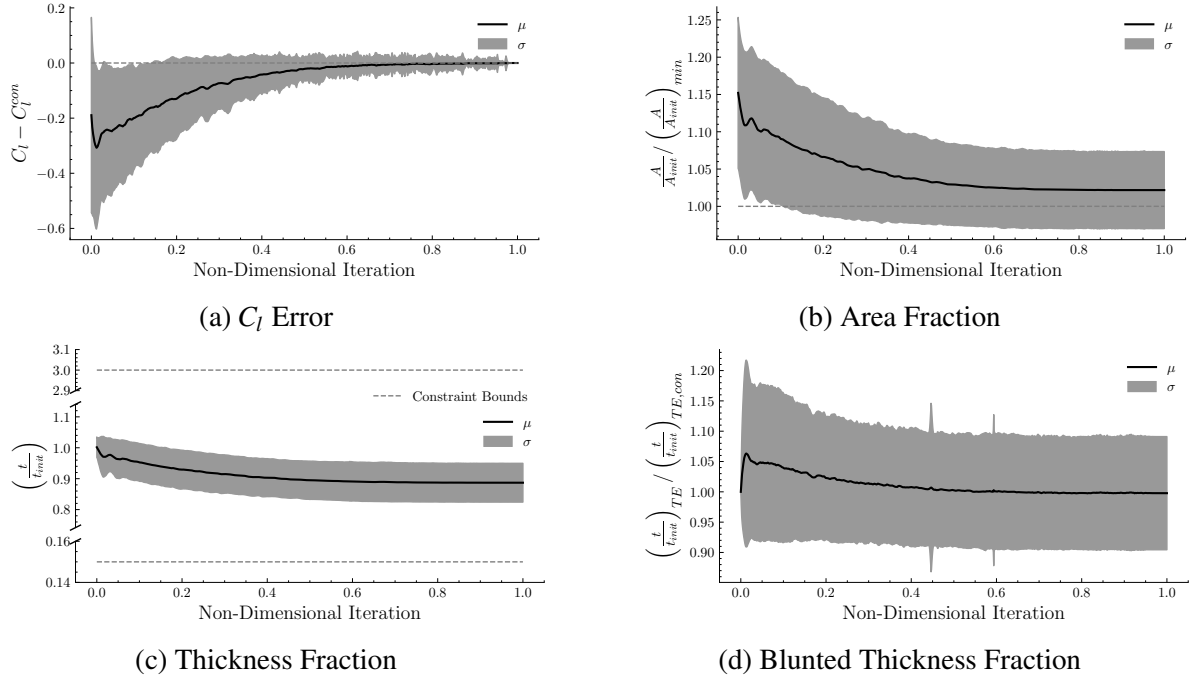
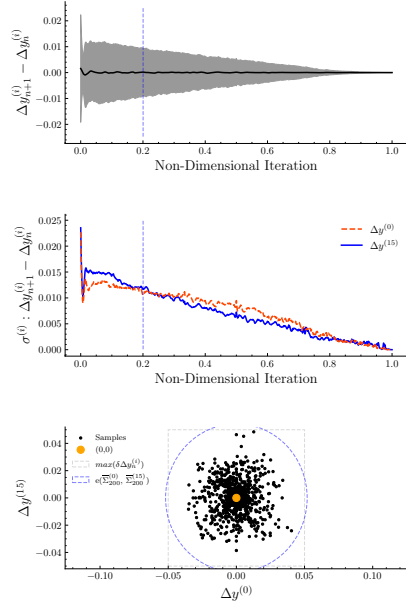
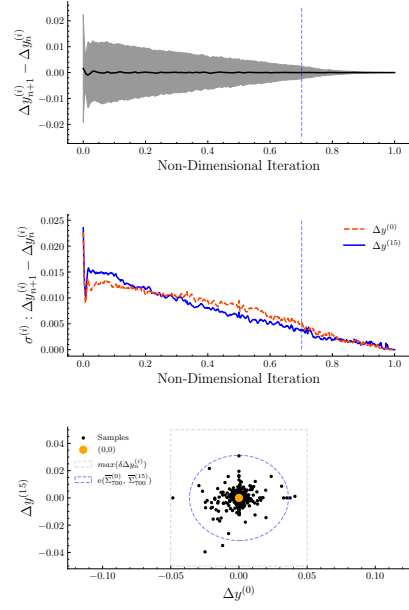


Figure 4.7: Aggregate Constraint Convergence

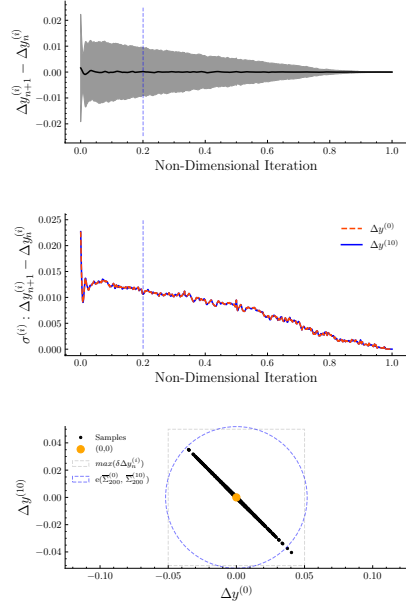
All constraints appear to decrease smoothly in the aggregate, and no abnormal behavior was observed. Additionally, we observed the thickness constraints were never violated in any of the optimization cases we ran. We also investigated how individual FFD points change per iteration, and whether they followed any sort of statistical pattern. We plotted the per iteration change in each FFD point against each other interactively as a function of optimization iteration. A static version of these results is presented in Figure 4.8.



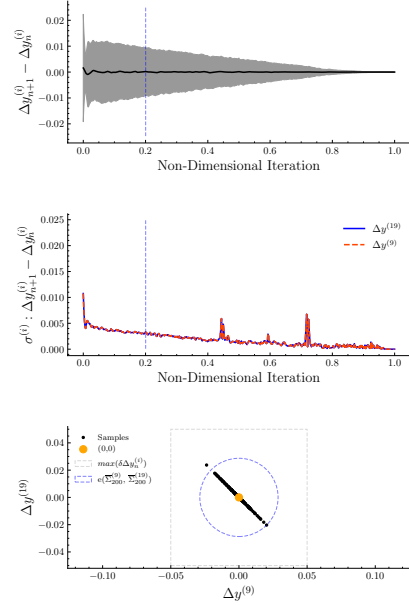
(a) First and 11th Points; 20% Along Trajectory



(b) First and 10th Points; 70% Along Trajectory



(c) Leading Edge Points; 20% Along Trajectory



(d) Trailing Edge Points; 20% Along Trajectory

Figure 4.8: Per-FFD Point Variation as a Function of Iteration

A significant takeaway from Figure 4.8 is that the relative change in FFD points appears to follow a relatively Gaussian-shaped distribution with a linearly decreasing variance and

a mean centered around zero. The difference between plots (a) and (b) in Figure 4.8 demonstrates how this Gaussian linearly shrinks as a function of trajectory. Finally, for validation purposes, we plotted the pairs of leading and trailing edge points (Figure 4.8 (c), (d)) which, as expected, follow a direct linear relationship, verifying the no-twist conditions we imposed.

Lastly, for reference, we present in more detail the standard deviations of the point changes for all 20 individual FFD points in Figure 4.9.

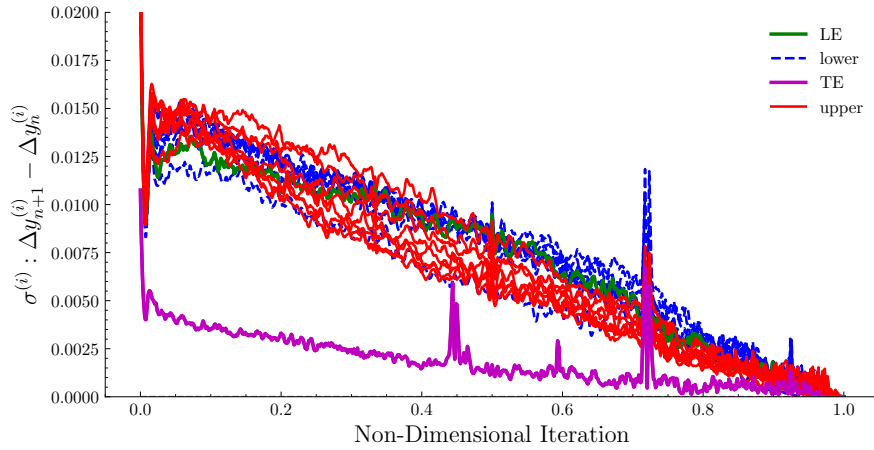


Figure 4.9: Standard Deviations in Variation for All 20 FFD Points

Based on these results, it appears the upper surface experiences slightly less variation in FFD point position than the lower surface—however, this difference is not major. We also note how variation in the trailing edge is extremely limited due to the blunted trailing edge thickness constraint we imposed.

Chapter 5: Generative Model Methodology

This chapter will describe the implementation of both models tested, our approach to running the warm start experiment, and finally our exploratory DDM inference efficiency study.

5.1 CEBGAN Modifications

In our work, we keep the majority of the CEBGAN model the same as it was originally presented. However, we do make a change to the architecture in order to accept initial airfoil design, x_0 , inputs. To do this we apply an auxiliary feedforward network which accepts the flattened x_0 input and outputs a set of latent variables which can be concatenated onto the inputs to the airfoil and angle of attack generators. The final model used has around 8 million parameters, and was trained with the same settings as specified in the original CEBGAN paper [2].

5.2 Aero-DDM Architecture

Note: This section is a modified version of the author’s original work as published in [6].

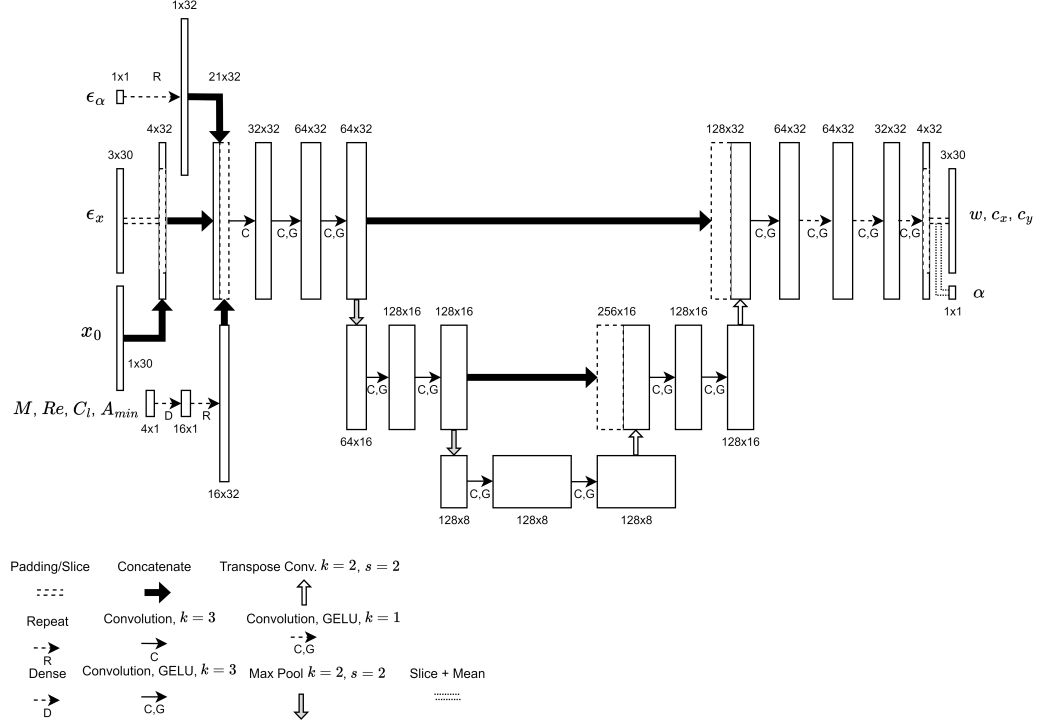


Figure 5.1: U-Net Architecture

Our model, Aero-DDM, takes on a standard U-Net architecture, which is commonly used in DDM implementations [15]. Our implementation differs in a few key aspects from the more standard image models. Most importantly, the model was trained in a latent space representing Bezier curve control points and weights (c_x , c_y , w), rather than in the raw coordinate space, following Chen *et al.* [4]. This latent space was achieved by pre-training an autoencoder on the training data. In this case, the training data also included all of the airfoils on the optimization trajectories. We use 32 Bezier control points, with 2 of the 32 control points being fixed at the trailing edge. Architecturally, 1-D convolutions with periodic padding were used in place of 2-D convolutions (which are better suited to pixel representations) to match the airfoil coordinate space. Conditioning on the flow parameters, (M , Re , C_l), the minimum area constraint, (A_{min}), and the initial airfoil design (x_0) was handled by a simple concatenation onto the latent space. Additionally, the scalar

noise variable for angle of attack, (ϵ_α) , was also repeated across the data dimension and concatenated in the same way. We note that since our work in [6], we have modified Aero-DDM to take the time-embedding, t_{emb} , (an embedding of t in the diffusion schedule) and concatenate it to the conditional inputs prior to passing it through a feed forward network, ϕ , which outputs a latent embedding, t_{latent} :

$$\phi(t_{emb}, c) = t_{latent}.$$

The model used was relatively small, with less than 500,000 trainable parameters, not including the pre-trained auto-encoder. Training wall time is about 10 minutes on an NVIDIA A1000 GPU, and sampling time was about 7s for the entire training set. This computational efficiency in sampling is on par with other generative models, such as GANs, implemented for similar generative airfoil design problems, but much better than these models in terms of training time required [2].

This output is then added prior to each ResNet concatenation in the U-Net. We found that doing this, along with reducing the learning rate on plateaus led to modest improvements in statistical benchmarks.

5.3 Bezier Autoencoder Validation

All models presented in this thesis make use of the same latent Bezier encoding of the airfoil coordinate space. In addition, the encoder used is a pre-trained model whose parameters are not changed during the training process of any of the generative models tested. As a result, the upper bound on model performance is decided by how accurately the original data can be reconstructed from the latent space. Therefore, it was important to validate the Bezier autoencoder’s performance. In Figure 5.2 we present the worst 6 reconstructions on the entire validation dataset (N=140) of optimized designs.

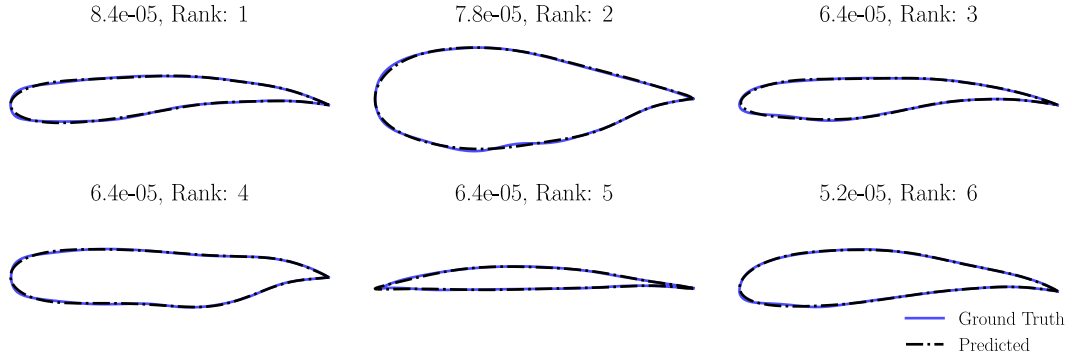


Figure 5.2: Top 6 Highest MSE Bezier Reconstructions in Validation Set

For the optimized dataset, the mean and standard deviation of the reconstruction error were $1.56\text{e-}05$ and $1.29\text{e-}05$ respectively. Airfoils shown in Figure 5.2 display the upper end of the variation in reconstruction error. Considering the designs in Figure 5.2 are the highest error cases, yet still visually match ground truth designs, we can be relatively confident in the robustness of the autoencoder’s reconstructions. In a sense, the mean reconstruction error of $1.56\text{E-}05$ represents a loose upper bound on the MSE of any model trained using the encoded control points and weights produced by the Bezier autoencoder model. In other words, all things being equal (and assuming no design space multi-modality), a theoretical generative model with the ability to perfectly predict optimal airfoil designs should expect a maximum achievable MSE between generated designs and the raw optimal airfoil coordinate data on the order of $1.56\text{e-}05$. Figure 5.3 shows the distribution of reconstruction error across the training and validation datasets.

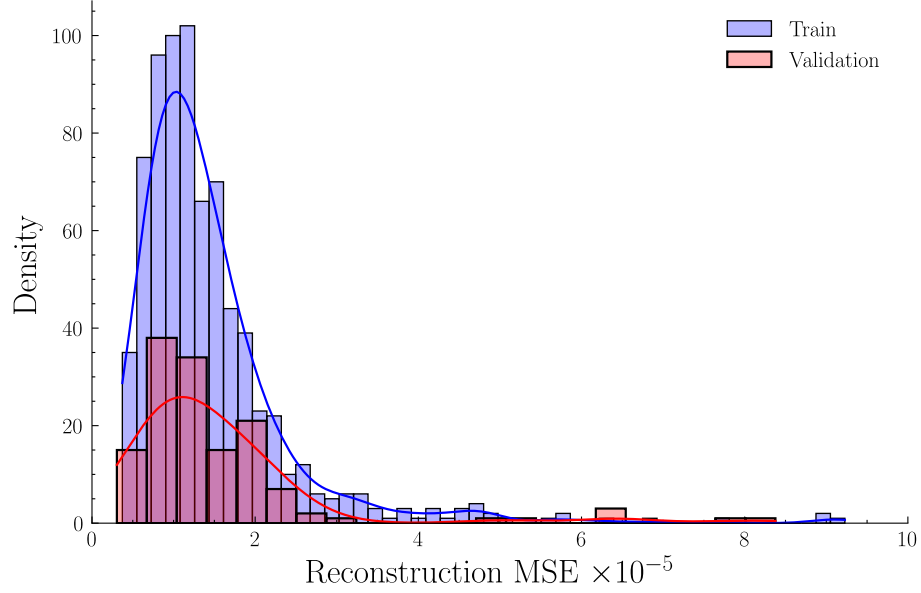


Figure 5.3: Distribution of MSE for Bezier Reconstructions

Solid blue and red lines represent Gaussian kernel density estimates for the training and validation sets respectively. The shaded regions are histogram bins.

The distribution in Figure 5.3 shows a slightly skewed error distribution, with a tail that extends to just less than $1\text{E-}4$. In addition, the validation data error distribution follows a very similar distribution to the training set, which indicates that overfitting is most likely not an issue.

We also note, that to avoid any accidental biases, we only made changes to the Bezier model based on validation data, and reserved the test data for the final diffusion and CEB-GAN model comparisons. Overall, the majority of airfoils in the dataset appear to be able to be encoded into the Bezier space with little loss in quality upon reconstruction.

5.4 Diffusion Parameters and Scheduling

We adopted a linear variance (β) schedule for the latent Bezier points ranging from $1\text{E-}4$ to $2\text{E-}2$, with a total of 1000 steps where the 1st step represents the lowest amount

of noise, and 1000 is the highest. This range was proposed by Ho *et al.* in their original diffusion paper [15]. A sample noised by the forward diffusion schedule for Aero-DDM is shown in Figure 5.4. Note we show latent Bezier control points (colored by weight) in the top row and the raw airfoil coordinates in the bottom row.

For α we chose a different schedule range of 1E-4 to 1E-1, as we determined higher levels of noise earlier in the schedule were necessary to avoid over-fitting, wherein the model would produce biased α values closer to the statistical mean of the dataset. This is possibly because early on in the schedule, α has little correlation to the actual airfoil shape, and so the loss can be minimized by simply predicting the statistical mean α value if not noised sufficiently—although further investigation may be required to fully validate this hypothesis.

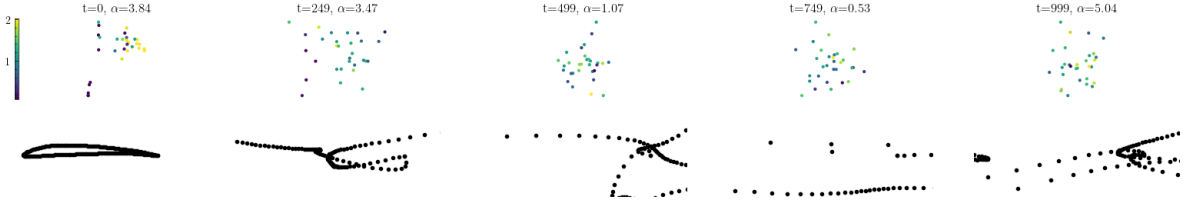


Figure 5.4: Forward Diffusion Schedule

5.5 Benchmarking and Warm Start Optimization

All models in our study are benchmarked on the test dataset (N=47), and generated 5 passes over this dataset for each model tested (N=235 airfoils generated and benchmarked per model variation). The only exception to this is the set of adjoint optimizations we ran using the original optimized designs as a starting point (discussed below), where we only ran (N=47) airfoils, since the adjoint optimizer is more or less deterministic in the designs generated given the same input conditions. Prior to evaluating any design, we interpolate the airfoil coordinates according to curvature to increase mesh quality and accuracy of

integrated quantities. This setup is the same for both the CFD and warm start benchmarks which we describe next.

Warm starting refers to the process by which a design is used as the initial state in an optimization process with the intent to accelerate convergence toward an optima. In our study we measure the warm start performance of our Aero-DDM (and its variants) and the CEBGAN model. We also run adjoint optimization cases starting from the original test set of optimized designs, x^* , to set a loose upper bound on the maximum expected performance. As opposed to the random UIUC designs, x_0 , used to collect the data, x^* should be the set of designs closest to the optima (or a local optima, if uni-modal), and therefore should also give the optimizer the largest warm-start efficiency advantage. Although starting the optimization with x^* may be expected to lead to the same x^* optima, in practice the results may vary slightly if the region around the optima is relatively flat, or the case is multimodal. As a result, comparisons to the optimization from x^* should partially control for some of this behavior. Important to note here, is that clearly x^* is a-priori knowledge of the optimal design, and thus this method is not computationally practical - it was only included as a point of comparison to the upper bound.

5.6 DDM Inference Efficiency Study

Throughout this thesis we run a study that attempts to determine whether reduced inference time, as measured by a reduction in diffusion sampling steps, can be obtained simply by changing the starting distribution from which the inference process is initialized. We tested variations on the baseline Aero-DDM model initialized from 3 different starting distributions; (1) pure Gaussian noise, $\mathcal{N}(0, I)$, as is also used to initialize the baseline Aero-DDM model, (2) CEBGAN generated designs, (3) x_0 , the initial airfoil designs used to generate the optimized test dataset. Here x_0 is the same initial (UIUC) airfoil design that

the adjoint solver used to start the optimization process. In

We conducted an ablation study measuring how many diffusion steps are required for reasonable distributional matching, as measured by airfoil coordinate MMD. The MMD ablation study was ran for 25 time step values between 1 and 350 steps, with each MMD evaluation involving the generation of 10 passes over the test dataset.

Chapter 6: Model Results

In this section, we present and discuss the following:

1. Qualitative comparisons of CEBGAN and Aero-DDM sample generations.
2. Statistical measures of how well the CEBGAN and Aero-DDM generated designs match the distributions of the dataset.
3. An inference efficiency ablation study, with discussion on the statistical performance of different initializations of the Aero-DDM inference process as a function of step number.
4. Performance of designs generated by CEBGAN, Aero-DDM, and the most promising variant from the inference efficiency ablation study in two areas:
 - (a) CFD-derived aerodynamic performance.
 - (b) Warm-start efficiency as measured by cumulative optimality-gap metrics.

6.1 Sample Generations

Input parameters from the held-out test set were used to statistically validate generations produced by each model. A grid of sample generations for Aero-DDM and the modified CEBGAN model is presented in Figure 6.2, along with the corresponding ground truth test data.

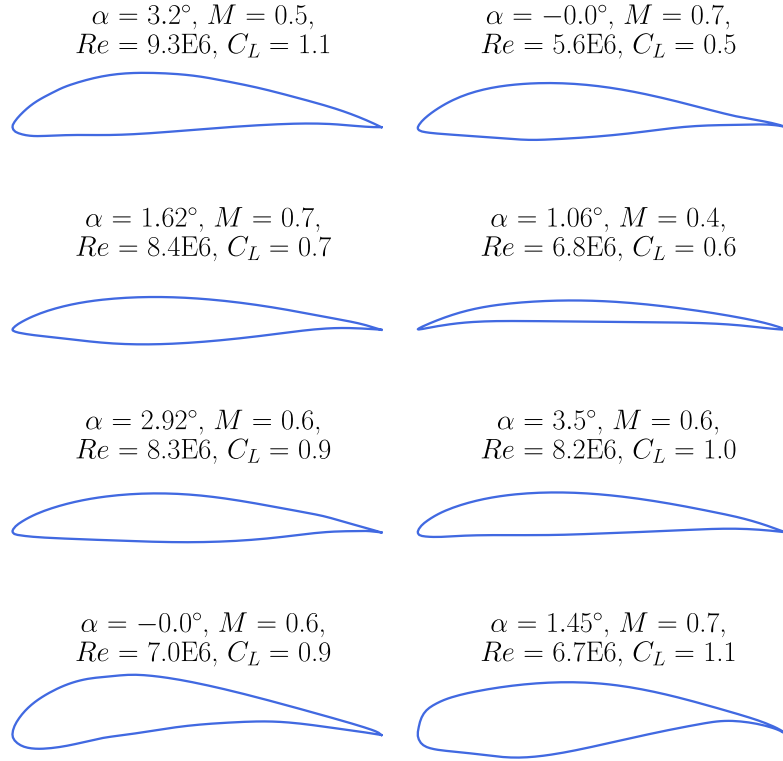


Figure 6.1: Sample Ground Truth Airfoils

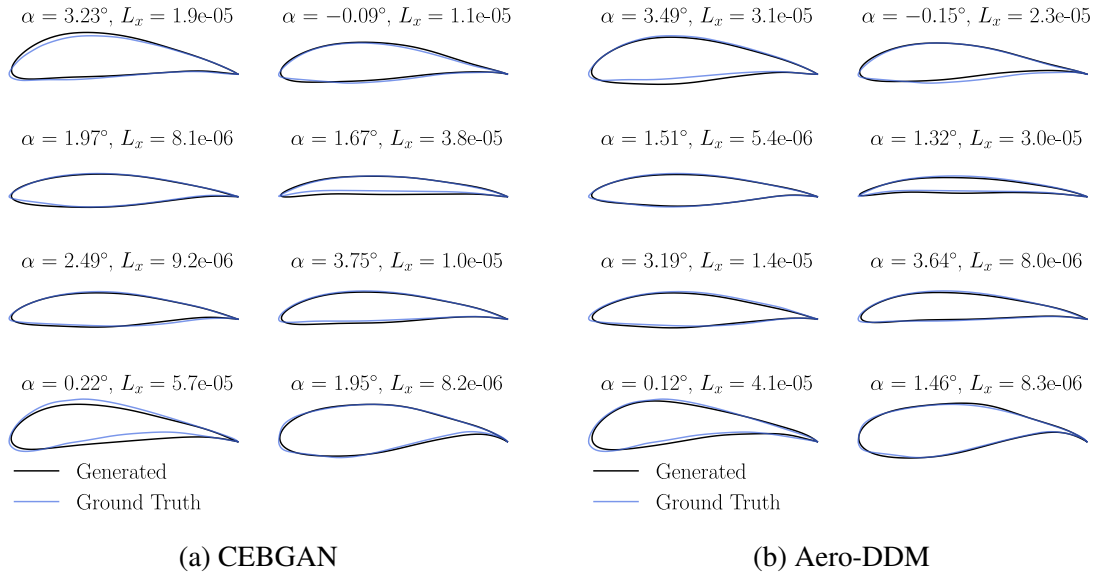


Figure 6.2: Sample Generations

L_x = Airfoil MSE

Both Aero-DDM and the CEBGAN model produced visually accurate generations. Even so, there are noticeable differences, particularly between for irregular airfoil shapes, whereby the CEBGAN performs worse than Aero-DDM, such as with the design in the lower left corner in Figures 6.2a and 6.2b. The other main observation is that the airfoils generated by CEBGAN also appear to have worse angle of attack predictions compared to Aero-DDM.

6.2 Statistical Validation

Model performance, as measured by MSE, MMD, and area constraint satisfaction are collated in Table 6.1.

Table 6.1: Statistical Benchmarks

Note: values to the right of the $\pm$ symbol indicate a 95 % confidence interval.

	MMD (Airfoil)	MMD (α)	MSE (Airfoil)	MSE (α)	Area Constraint MSE
Aero-DDM	0.063 $\pm$ 4.3E-04	0.060 $\pm$ 0.0028	1.55E-05 $\pm$ 4.7E-07	0.067 $\pm$ 0.01	2.34E-06 $\pm$ 4.1E-07
CEBGAN [2]	0.072 $\pm$ 3.4E-05	0.106 $\pm$ 0.0010	2.16E-05 $\pm$ 4.2E-08	0.281 $\pm$ 0.003	2.88E-06 $\pm$ 5.5E-08

The results in Table 6.1 appear to confirm the qualitative differences observed in Figure 6.2, with Aero-DDM having marginally better performance in MMD, and modestly better performance in angle of attack. The observed difference in angle of attack may have more to do with the fact that the CEBGAN architecture uses a separate network to generate angle of attack predictions that is not internally dependent on the airfoil-generating portion of the network [2]. In contrast, the angle of attack predictions produced by Aero-DDM are coupled to the airfoil latent coordinates prediction, with the result being the average of one of the convolution channel outputs.

Another observation is that the Area Constraint MSE is relatively similar for both models, and in general performance on the order of $1\text{E-}6$ is a strong indicator that both models

have learned to generate designs which respect this constraint either through the explicit A_{min} input, or implicitly through x_0 . A promising avenue for future work could be the study of how access to A_{min} or x_0 affects area constraint satisfaction.

6.3 DDM Inference Ablation Study

After statistically benchmarking the CEBGAN and Aero-DDM, we conducted an ablation study on airfoil MMD values for the three different initialized variants of Aero-DDM as described earlier. We briefly address these results here, as they explain our focus on the x_0 initialized Aero-DDM variant in all subsequent performance benchmarks.

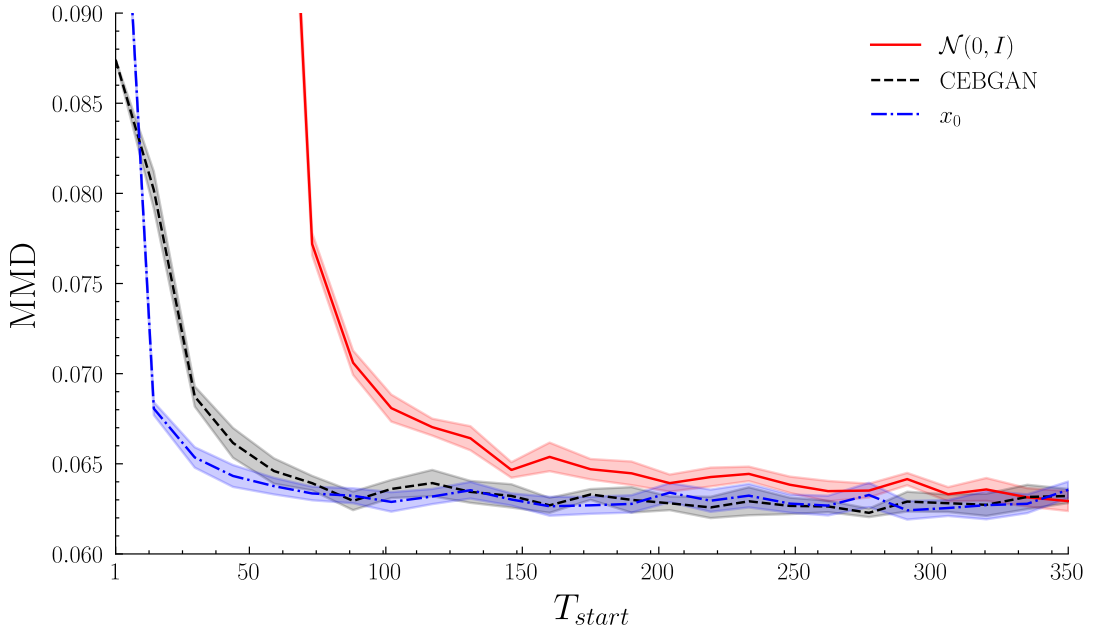


Figure 6.3: Inference Steps Ablation Study

Figure 6.3 displays MMD as a function of starting time-step for each scenario. Lower values of T_{start} indicate less denoising steps (i.e more efficient inference). In addition, lower values also mean a lower assumed starting variance β . This is a key point, as lower starting variances mean less cumulative change can be effected on the initial design by the model.

An example of the effect of T_{start} on design quality can be found in appendix C.

Results from Figure 6.3 indicate that all initialization approaches tend to converge on the original Aero-DDM MMD value, and that the difference is negligible after inferences with 350 steps or more. This is an interesting result for the $\mathcal{N}(0, \mathbf{I})$ started model, as it appears even an unadulterated version of Aero-DDM that works with pure noise can achieve the same level of accuracy as the base model in only 350 steps. The most interesting result however, is that the x_0 initialized variant produces a similar MMD to the base model in only 50 steps. This represents a 95 % reduction in steps needed compared to the original 1000 step, $\mathcal{N}(0, \mathbf{I})$ started inference process, but considering the $\mathcal{N}(0, \mathbf{I})$ started approach also achieved a similar MMD result at 350 steps, a fairer estimate would be a decrease of $\sim 86\%$. In addition, the x_0 case appears to obtain lower MMD values in less steps than even the CEBGAN initialized version of Aero-DDM. This is surprising because the airfoils produced by the CEBGAN share a much higher distributional similarity to the ground-truth optimal airfoils compared to the initial (x_0) designs. Figure 6.4 presents some of the different initializations of the Aero-DDM inference process at $T_{\text{start}} = 50$.

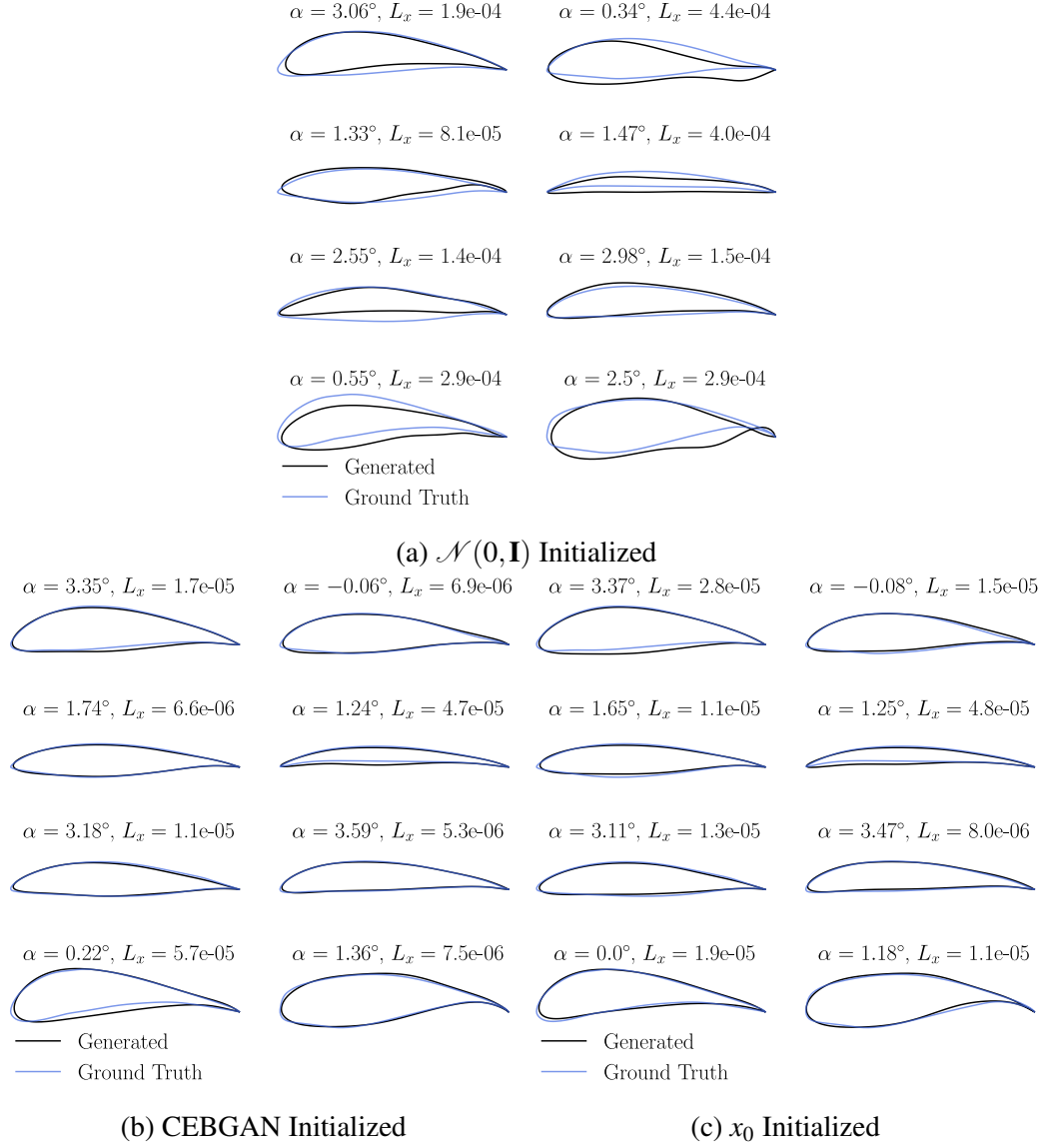


Figure 6.4: Sample Generations for Different Aero-DDM Initializations, $T_{start} = 50$

$$L_x = \text{Airfoil MSE}$$

While the $\mathcal{N}(0, \mathbf{I})$ initialized generations in Figure 6.4a are clearly inferior to the other initializations in terms of quality, some more interesting observations can be made about the other two cases. Firstly, while the CEBGAN initialized samples appear to mimic the generations of the original CEBGAN in Figure 6.2a as expected, there also appears to be a slight improvement in angle of attack. What is more surprising, however, is that the

angle of attack predictions by the x_0 initialized Aero-DDM actually demonstrate a modest improvement compared to even the baseline Aero-DDM (see Figure 6.2b, $T_{start} = 1000$) across all but one sample. Since the predicted airfoil shapes of the x_0 initialized Aero-DDM are mostly the same as the baseline, it is reasonable to conclude that any differences in performance observed can be attributed to this angle of attack prediction improvement.

We leave a more rigorous study of the mechanism behind this difference as future work. The impact of these results caused us to narrow our focus on the specific case of the x_0 initialized Aero-DDM model, which appears to be the most efficient and best performing variation. Throughout the rest of the results, we will present performance benchmarks for the x_0 initialized variant for both $T_{start} = 50$ and 100 steps.

6.4 Aerodynamic Performance Benchmarks

In addition to our statistical analysis, we benchmarked the generated designs using the same ADflow RANS CFD solver and meshing procedure used to create the dataset. These results are collated in Table 6.2. In general, we observed that most benchmarked cases succeeded, with 233 successes for the CEBGAN model and 211 for Aero-DDM. However, as some of these failures were related to technical issues on the HPC system used, rather than as a result of solver convergence issues, it is difficult to make any conclusions based on failure rate alone.

We also note some variation in our results as compared to our previous work in [6], which can be attributed to using a different subset of the test set. In that study, a higher proportion of simulations failed and therefore had to be pruned from the test set as well in order to keep our comparisons to the same subset of airfoils.

In addition to the benchmarked models, Table 6.2 includes performance measures of the unoptimized airfoils from our original dataset as a point of reference.

Table 6.2: Aerodynamic Performance Benchmarks

Note: values to the right of the $\pm$ symbol indicate a 95 % confidence interval.

	$ C_d - C_d^* $ (Counts)	$ C_l - C_l^{con} $	$\text{IOG}\left(\frac{C_l}{C_d}\right)$
Aero-DDM	9.1 ± 2.6	0.0292 ± 0.0039	5.0E-4 $\pm$ 1.0E-4
CEBGAN [2]	18.4 ± 5.2	0.0509 ± 0.0064	$6.8\text{E-}4 \pm 1.4\text{E-}4$
Aero-DDM ($x_0; T = 50$)	9.1 ± 2.4	0.0260 $\pm$ 0.0031	$5.4\text{E-}4 \pm 1.1\text{E-}4$
Aero-DDM ($x_0; T = 100$)	8.3 $\pm$ 2.2	0.0287 ± 0.0040	$5.1\text{E-}4 \pm 1.0\text{E-}4$
Unoptimized	37.3 ± 21.3	0.2740 ± 0.0570	$23.2\text{E-}4 \pm 4.5\text{E-}4$

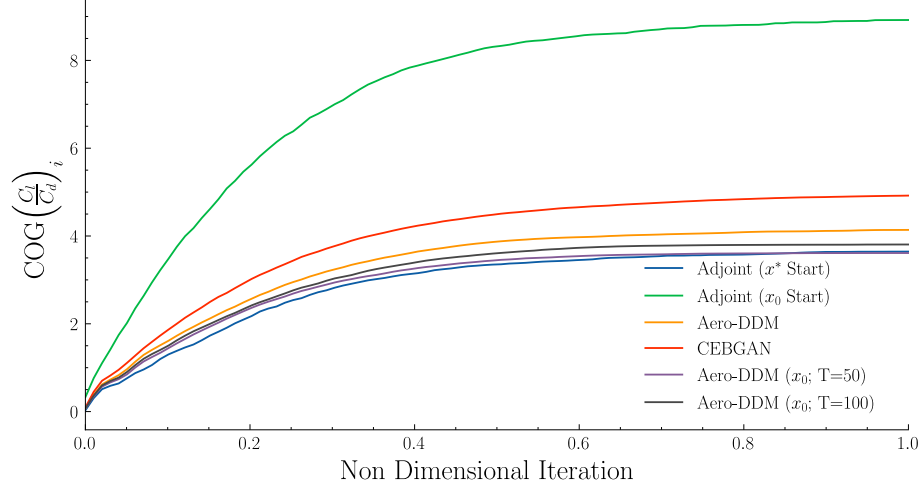
Results reported in Table 6.2 indicate that compared to CEBGAN, Aero-DDM produces a $\sim 51\%$ lower optimal C_d gap, $\sim 43\%$ lower C_l constraint satisfaction and $\sim 26\%$ lower instantaneous optimality gap (IOG) for $\frac{C_l}{C_d}$. All models benchmarked exhibit an improvement over the unoptimized data.

Taking into consideration the models used in the inference efficiency study, we note that the 50-step x_0 initialized Aero-DDM produces a greater reduction in C_d gap and a similar IOG to that of the baseline Aero-DDM. There appears to be little difference between the 50 and 100-step variants of the x_0 initialized approach across all metrics. In addition, none of these results indicate significant performance degradation costs associated with the x_0 approaches.

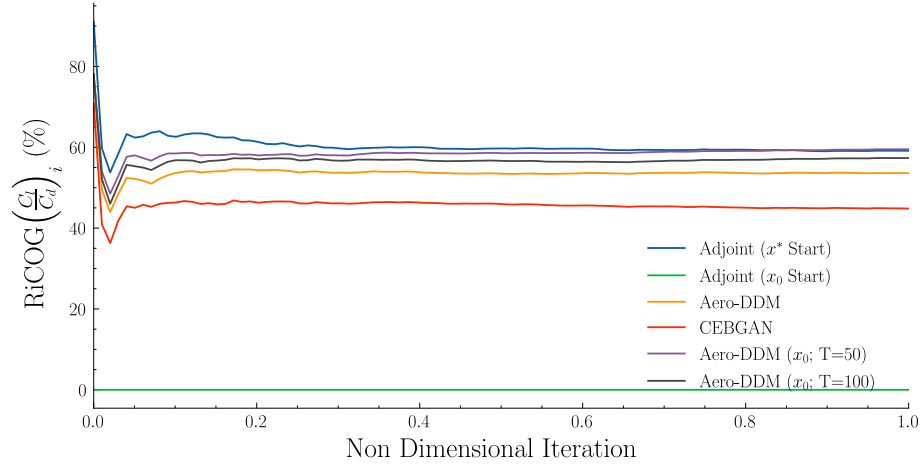
6.5 Warm Start Performance

Figure 6.5a presents the COG and RiCOG optimality gap metrics as a function of non-dimensionalized iteration. We have also included comparisons to the original adjoint dataset (Adjoint x_0 start), as well as the results obtained from running the adjoint solver

continuing from the optimal airfoil designs from the original adjoint data (Adjoint x^* start).



(a) Cumulative Optimality Gap



(b) Reduction in Cumulative Optimality Gap

Figure 6.5: Warm Start Performance

Before analyzing the relative performance of the models, we can make some comments on the results of the study as a whole. The first observation is that all models tested—as well as the adjoint continuation cases—experience a sharp decrease in RiCOG (Figure 6.5b) at the beginning of the optimization trajectory. This is most likely due in part to properties of the SLSQP solver that cause it to test the bounds of the constrained optimization problem to guarantee some level of global convergence [19]. In addition, it is important to note

that in all warm start cases tested, including the adjoint optimal start case, the final design often produced designs with slightly improved performance over the original adjoint optimized designs. This was also noticed by Chen *et al.* in [2], and may indicate that the optimal designs in the original dataset are local optima, and that the act of re-running the optimization allowed for the solver to fine-tune the result within a tighter radius of convergence—although this hypothesis needs to be tested in future work. Even so, our inclusion of the adjoint optimized results starting from the optimal airfoil designs should bound the uncertainty related to the local optima issue.

Table 6.3 reports the optimality gap metric results from our warm start experiment. Both the COG and RiCOG values listed are the final values at the end of the optimization trajectory.

Table 6.3: Warm Start Performance Benchmarks

Note: values to the right of the $\pm$ symbol indicate a 95 % confidence interval.

	$\text{COG}\left(\frac{C_l}{C_d}\right)_N$	$\text{RiCOG}\left(\frac{C_l}{C_d}\right)_N$ (%)
Aero-DDM	4.20 ± 0.66	53.60 ± 6.90
CEBGAN [2]	4.96 ± 0.76	44.84 ± 4.77
Aero-DDM (x_0 ; T=50)	3.62 ± 0.49	59.51 ± 8.87
Aero-DDM (x_0 ; T=100)	3.82 ± 0.53	57.33 ± 8.59
Adjoint (x^* Start)	3.71 ± 1.14	59.14 ± 18.67
Adjoint (x_0 Start)	9.03 ± 2.07	0 (Defines RiCOG)

Consistent with our expectations from the CFD benchmark performance, we observe that Aero-DDM produces a lower COG and higher RiCOG than the CEBGAN model. Both of these results imply less wasted optimizer effort when Aero-DDM generations are used

in warm start scenarios. Even so, the difference between the two models is relatively subtle ($\sim 20\%$ for RiCOG, $\sim 15\%$ for COG), compared to the difference in IOG. This may indicate that the optimizer cannot take full advantage of an airfoil close to the optimal design, or that the airfoil designs produced by the Aero-DDM may have qualities (such as non-optimal C_l satisfaction) that require additional effort to resolve. One of the more surprising results, is that the x_0 initialized, $T=50$ sample efficient Aero-DDM actually shows modest improvement in both RiCOG and COG over the original Aero-DDM. The reasoning behind this is unclear, and is an area for future work, but the result is consistent as a function of T_{start} , with the $T=100$ variant also displaying improvement over the original Aero-DDM. Finally, we note that, as expected, the adjoint x^* continuation case shows, with exception of the $T=50$ x_0 initialized Aero-DDM, the lowest and highest COG and RiCOG values. Although the $T=50$ x_0 initialized Aero-DDM performs better, we also note the higher uncertainty bounds on the adjoint x^* continuation case which could easily wipe out this gap, considering the gap in mean value between the two is already relatively small.

Chapter 7: Conclusions

In this thesis, we have presented several key contributions which will help inform the current use, and future study, of generative models for aerodynamic design purposes. First, we have presented a novel dataset of 935 optimized airfoils obtained from a diversely sampled design space. Second, we demonstrated success using our latent denoising diffusion model, Aero-DDM, which features a custom U-Net architecture and simple conditioning on the initial airfoil design and flow parameters. Next, we compared Aero-DDMs performance to an optimal-transport based GAN model, CEBGAN, and although both models produced reasonably performant designs, we observe that Aero-DDM produces designs which better match the data-distribution, and which produce lower drag values in aggregate based on aerodynamic performance benchmarks. We have tested the ability of both models to accelerate convergence of an adjoint solver, when the generated designs are used to warm-start optimizations. Based on these warm start results, we also observe that Aero-DDM produces a higher reduction in cumulative optimality gap, and is therefore is the more efficient choice for these scenarios—although we acknowledge this difference is not major. Finally, considering the high inference cost associated with diffusion models compared to GANs, we experimented with accelerating sampling by having Aero-DDM begin the sampling process with a distribution much closer than pure noise to the optimal data distribution. The results of this experiment appear to demonstrate that by simply starting the inference process from the initial design distribution, sampling efficiency can be improved by $\sim 86\%$ compared to an optimized version of the baseline inference pro-

cess, without meaningful degradation in performance. In addition the designs generated through this approach appear to have better warm-start performance than even the baseline Aero-DDM model.

7.1 Future Work

This section presents improvements to the methodology in this thesis, as well as various directions for future studies. Here we summarize future work before describing our suggestions in more detail. First, improvements can be made to data collection process, including validating mesh quality through a refinement study and comparing the results for different optimizers. In addition, changes to the optimization problem, such as increasing FFD deformation allowances and modifying constraints, could be made to tackle more complex design problems. Next, investigations into model inputs, such as the current inclusion of the initial design, and the effect of the Bezier latent space could provide useful insight into the performance and generalizability of Aero-DDM. Additionally, measurements of out-of-distribution performance of the models on a larger flight window could be done to test the robustness of the generated designs. Finally, an exploration into the mechanism behind the improved sample efficiency we observed when using the initial design distribution would help further optimize model performance, and may provide insight into how widely applicable the method is.

In general there are many ways the quality of the dataset could be improved upon if the collection process were to be repeated. Firstly, although we believe the optimized airfoils we generated were of high quality, and the setup we used was sufficient in terms of replicating the general trends in optimal airfoil shape as a function of flow parameters, there would be a great benefit to conducting a rigorous mesh-refinement study at a variety of flow conditions and initializations. This could take the form of a h-refinement (grid-spacing re-

finement) volume mesh coarsening study, with the results aggregated across a sparse subset of the design space. Secondly, the SLSQP optimizer we used may not be the best choice for measuring warm start performance, given the global convergence properties it attempts to ensure, which may inadvertently wash out some of the benefits of starting closer to the optimal design. In addition, other nonlinear optimizers, such as SNOPT exhibit much smoother, stable, optimization trajectories, which may make analysis of these trajectories much easier [10]. Finally, future work should explore changes to the optimization problem itself, such as increasing the amount of deformation (Δy_i) the optimizer is allowed to make to the FFD cage. This may lead to more multi-modal behaviour and represents a more difficult, realistic, design problem. To do this, robustness improving techniques such as FFD adaptation may have to be used to avoid convergence related failures [14]. Additionally, changing C_l to an inequality constraint from an equality constraint would help make $\frac{C_l}{C_d}$ a much more reliable indicator of performance.

Certain changes to the model inputs themselves could be interesting explore. For example, removing x_0 , the initial design, from the set of inputs could reveal how much of an impact it has on the generated design. A high dependency on x_0 , may also prompt further study into what degree the dataset generated by the adjoint optimizer itself is multi-modal depending on initial conditions. Secondly, further investigations into the Bezier latent space could be conducted. Such investigations may explore the minimum number of control points needed to produce accurate reconstructions, which would be important to understand if the latent model were to be extended to a higher-dimensional problem. It may also be possible to explore how to cluster the control points to create better reconstructions of sharper leading edge profiles, as the current autoencoder implementation struggles with this. Future work could also explore alternatives to Bezier representations. In particular, parameterization using Class-Shape Transformations (CST) may be of interest, since CST design variables may be more interpretable, exclude many infeasible designs, and provide

a better leading edge representation. Apart from studying the latent coordinate representation, a sample efficiency study of the model could be undertaken to measure the effect of the number of training samples on generation quality. Such a study could reveal which model deals better in situations with limited data.

In this thesis, we used test data that was still inside the LHS sampled flight window. A future study could gather data outside this range and use it to measure model robustness. Another approach could be to set the initial design, minimum area constraint, and C_l to a constant value, and see how well the model extrapolates in different Mach and Reynolds regimes, as compared to the adjoint-optimized designs. Based on preliminary results, which are not presented in this thesis, sonic and supersonic performance may be the most difficult regions to extrapolate to, and therefore could be a good starting point for any such future study.

Finally, while the inference study we presented led to surprising results, we did attempt to robustly study the underlying mechanism behind the increase in efficiency we observed. Future work could address this by systematically studying the distributional differences in the designs produced with the x_0 initialized approach and the baseline Aero-DDM. An additional, straightforward, approach could be to gather the same CFD and warm-start benchmarks, but for a greater number and range of T_{start} values, instead of the $T_{\text{start}} = 50, 100$ used in this work. Specifically, using T_{start} values less than 50 and above 100 may help determine for what value an optimally efficient T_{start} exists. It would also be useful to observe whether efficiency can be further improved applying one of the many suggested modifications to diffusion sampling algorithms which exist in literature. If the inference efficiency benefits do not compound with changes to the sampling algorithm, then future work could explore which method produces higher efficiency gains, and the scenarios in which the x_0 approach may be more or less appropriate.

Appendix A: Other Geometric Parameterization Methods

Although this thesis focuses on Bezier parameterizations, further research may benefit from experimenting with using other methods. Here we briefly cover some of the alternatives. Generally speaking, there is a tradeoff between how accurately these methods can reconstruct a diverse set of airfoils, and the degree to which infeasible designs are excluded through restrictions to the method itself. For example, highly generalizable methods such as singular value decomposition (SVD) have been shown to provide the most efficient coverage of the design space [25]. Despite this, SVD also tends to create many infeasible designs with crossover intersections and high curvature [3]. Other methods construct airfoil geometries using combinations of basis functions, which can be modified to enforce designs with properties such as a minimum thickness, smoothness, and continuity. Some examples include Hicks-Henne bump functions, which add parameterized sine functions to an initial airfoil design, and class shape transformations (CSTs), which modify a 'shape function' composed of Bernstein polynomials. CST stands out in particular as it does not require an initial design (i.e it is a constructive, not a deformative, method), can be used to create smooth shapes with no crossover, and can be modified to create highly accurate reconstructions of the leading edge [25]. Although the constructive Bezier method we use in this thesis can still suffer from some of these issues, the Bezier method is able to capture a very diverse set of airfoil designs with relatively small amounts of coordinate reconstruction loss. We leave a study of whether the feasibility advantages of methods such as CST can outweigh any generalizability issues they may have, if such tradeoffs exist at all.

Appendix B: Example Optimization Trajectory Plots for a Single Case

Here we present the optimization history of the objective and constraints for a random individual case in the dataset as a point of comparison to the aggregate quantities presented in chapter 4. The initial and final designs are presented in Figure B.1, and the histories are presented in Figures B.2 and B.3.

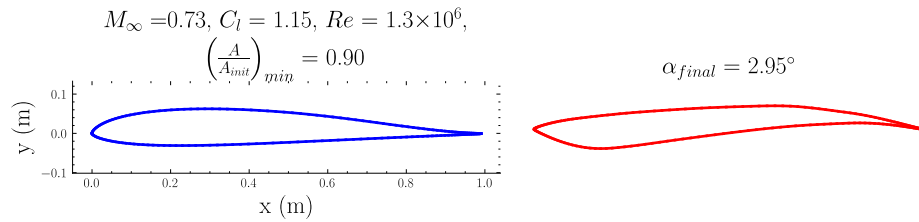


Figure B.1: Initial and Optimized Airfoil for Example Case

From left to right; unoptimized, optimized

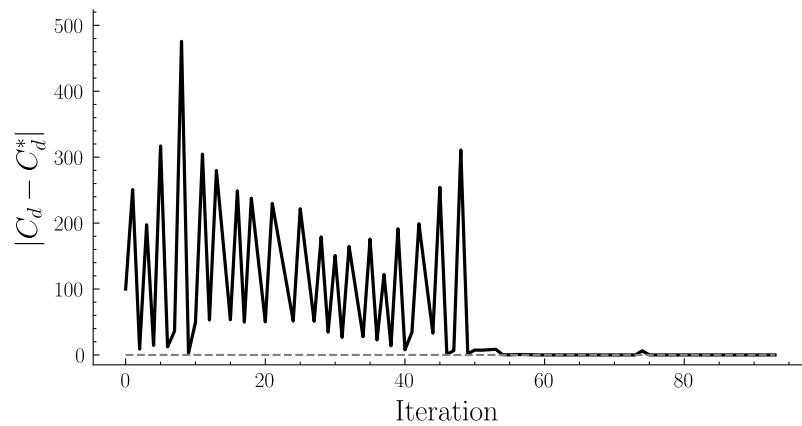
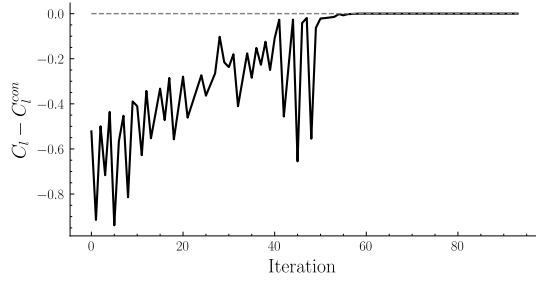
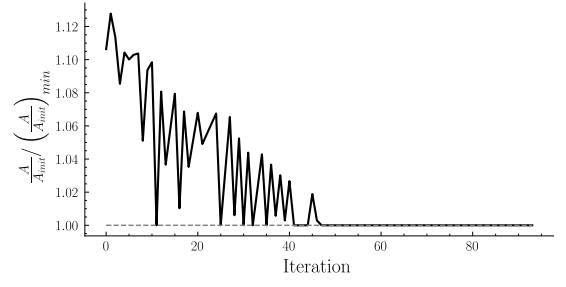


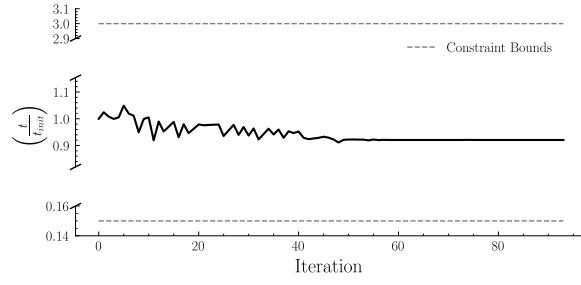
Figure B.2: C_d Convergence for Example Case



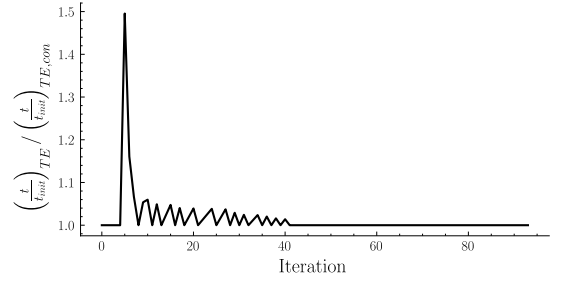
(a) C_l Error



(b) Area Fraction



(c) Thickness Fraction



(d) Blunted Thickness Fraction

Figure B.3: Constraint Convergence for Example Case

Appendix C: Effect of T_{start} on Designs

For illustrative purposes, Figure C.1 shows an example of the effect T_{start} has on the generated design for the x_0 initialized method.

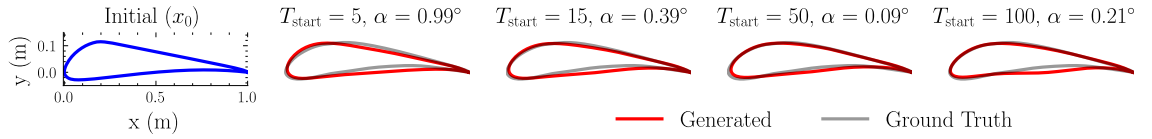


Figure C.1: Example Effect of T_{start} on x_0 Initialized Designs

Here, the ground truth is the adjoint optimized design. Figure C.1 demonstrates the main concept of how lower values of T_{start} collapse the solution to the original initial design, x_0 , as for these values of T_{start} , the model has much less ability to edit the input.

Bibliography

- [1] Gabriel Achour, Woong Je Sung, Olivia J Pinon-Fischer, and Dimitri N Mavris. Development of a conditional generative adversarial network for airfoil shape optimization. In *AIAA Scitech 2020 Forum*, page 2261, 2020.
- [2] Qiuyi Chen, Jun Wang, Phillip Pope, Wei (Wayne) Chen, and Mark Fuge. Inverse Design of Two-Dimensional Airfoils Using Conditional Generative Models and Surrogate Log-Likelihoods. *Journal of Mechanical Design*, 144(2), 12 2021. 021712.
- [3] Wei Chen, Kevin Chiu, and Mark D Fuge. Airfoil design parameterization and optimization using bézier generative adversarial networks. *AIAA journal*, 58(11):4723–4735, 2020.
- [4] Wei Chen, Kevin Chiu, and Mark D. Fuge. Airfoil design parameterization and optimization using bézier generative adversarial networks. *AIAA Journal*, 58(11):4723–4735, 2020.
- [5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.
- [6] Cashen Diniz and Mark Fuge. Optimizing diffusion to diffuse optimal designs. In *AIAA SCITECH 2024 Forum*, page 2013, 2024.
- [7] Xiaosong Du, Ping He, and Joaquim RRA Martins. A b-spline-based generative adversarial network model for fast interactive airfoil aerodynamic optimization. In *AIAA Scitech 2020 Forum*, page 2128, 2020.
- [8] MHF El-Gohary and MYM Ahmed. Investigating the design parameters of a diamond-shaped supersonic airfoil. In *International Conference on Aerospace Sciences and Aviation Technology*, number AEROSPACE SCIENCES & AVIATION TECHNOLOGY, ASAT-15–May 28-30, 2013, pages 1–18. The Military Technical College, 2013.

- [9] Sayan Ghosh, Govinda A Padmanabha, Cheng Peng, Steven Atkinson, Valeria Andreoli, Piyush Pandita, Thomas Vandeputte, Nicholas Zabaras, Liping Wang, and Genghis Khan. Pro-ml ideas: A probabilistic framework for explicit inverse design using invertible neural network. In *AIAA Scitech 2021 Forum*, page 0465, 2021.
- [10] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
- [11] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.
- [12] Hannah M. Hajdik, Anil Yildirim, Neil Wu, Benjamin J. Brelje, Sabet Seraj, Marco Mangano, Joshua L. Anibal, Eirikur Jonsson, Eytan J. Adler, Charles A. Mader, Gaetan K. W. Kenway, and Joaquim R. R. A. Martins. pyGeo: A geometry package for multidisciplinary design optimization. *Journal of Open Source Software*, 8(87):5319, 2023.
- [13] GM Harshvardhan, Mahendra Kumar Gourisaria, Manjusha Pandey, and Siddharth Swarup Rautaray. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285, 2020.
- [14] Xiaolong He, Jichao Li, Charles A. Mader, Anil Yildirim, and Joaquim R. R. A. Martins. Robust aerodynamic shape optimization—from a circle to an airfoil. *Aerospace Science and Technology*, 87:48–61, April 2019.
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [16] Antony Jameson. Aerodynamic shape optimization using the adjoint method. *Lectures at the Von Karman Institute, Brussels*, 2003.
- [17] Gaetan K. W. Kenway, Charles A. Mader, Ping He, and Joaquim R. R. A. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110:100542, October 2019.
- [18] Gaetan K.W. Kenway, Charles A. Mader, Ping He, and Joaquim R.R.A. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110:100542, 2019.
- [19] Dieter Kraft. A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1988.
- [20] Jichao Li, Mohamed Amine Bouhlel, and Joaquim R. R. A. Martins. Data-based approach for fast airfoil analysis and optimization. *AIAA Journal*, 57(2):581–596, 2019.

- [21] Jichao Li, Xiaosong Du, and Joaquim RRA Martins. Machine learning in aerodynamic shape optimization. *Progress in Aerospace Sciences*, 134:100849, 2022.
- [22] Wu Li, Steven Krist, and Richard Campbell. Transonic airfoil shape optimization in preliminary design environment. *Journal of aircraft*, 43(3):639–651, 2006.
- [23] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.
- [24] Charles A. Mader, Gaetan K. W. Kenway, Anil Yildirim, and Joaquim R. R. A. Martins. ADflow—an open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. *Journal of Aerospace Information Systems*, 2020.
- [25] Dominic A Masters, Nigel J Taylor, TCS Rendall, Christian B Allen, and Daniel J Poole. Geometric comparison of aerofoil shape parameterization methods. *AIAA journal*, 55(5):1575–1589, 2017.
- [26] François Mazé and Faez Ahmed. Diffusion models beat gans on topology optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.
- [27] Applied Aerodynamics Group Michael Selig, University of Illinois Urbana-Champaign. Uiuc dataset, 2023. data retrieved from , https://m-selig.ae.illinois.edu/ads/coord_database.html.
- [28] Jeremy Morton, Mykel J. Kochenderfer, and Freddie D. Witherden. Parameter-conditioned sequential generative modeling of fluid flows. *AIAA Journal*, 59(3):825–841, 2021.
- [29] Jitesh H. Panchal, Mark Fuge, Ying Liu, Samy Missoum, and Conrad Tucker. Special Issue: Machine Learning for Engineering Design. *Journal of Mechanical Design*, 141(11), 10 2019. 110301.
- [30] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [31] Lyle Regenwetter, Amin Heyrani Nobari, and Faez Ahmed. Deep Generative Models in Engineering Design: A Review. *Journal of Mechanical Design*, 144(7), 03 2022. 071704.
- [32] Melissa Rivers and Ashley Dittberner. Experimental investigations of the nasa common research model in the nasa langley national transonic facility and nasa ames 11-ft transonic wind tunnel. In *49th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 1126, 2011.

- [33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.
- [34] Ney Secco, Gaetan K. W. Kenway, Ping He, Charles A. Mader, and Joaquim R. R. A. Martins. Efficient mesh generation and deformation for aerodynamic shape optimization. *AIAA Journal*, 2021.
- [35] Ney Secco, Gaetan K. W. Kenway, Ping He, Charles A. Mader, and Joaquim R. R. A. Martins. Efficient mesh generation and deformation for aerodynamic shape optimization. *AIAA Journal*, 2021.
- [36] Shaun N Skinner and Hossein Zare-Behtash. State-of-the-art in aerodynamic shape optimisation methods. *Applied Soft Computing*, 62:933–962, 2018.
- [37] Neil Wu, Gaetan Kenway, Charles A. Mader, John Jasa, and Joaquim R. R. A. Martins. pyoptsparse: A python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54):2564, 2020.
- [38] Anil Yildirim, Gaetan KW Kenway, Charles A Mader, and Joaquim RRA Martins. A jacobian-free approximate newton–krylov startup strategy for rans simulations. *Journal of Computational Physics*, 397:108741, 2019.
- [39] Kazuo Yonekura and Katsuyuki Suzuki. Data-driven design exploration method using conditional variational autoencoder for airfoil design. *Structural and Multidisciplinary Optimization*, 64(2):613–624, 2021.