Deep Learning Techniques for Aerodynamic Wing Shape Optimization

Alfiyandy Hariansyah[1,2]

[1] *Department of Aerospace Engineering, Tohoku University, Sendai, Japan*
[2] *Institute of Fluid Science, Tohoku University, Sendai, Japan*

Boeing Higher Education Program Seminar
Year 2021-2022 Performance Report

Sendai, February 16, 2023

Contents

Why deep learning in wing design?

Aerodynamic Wing Shape Optimization

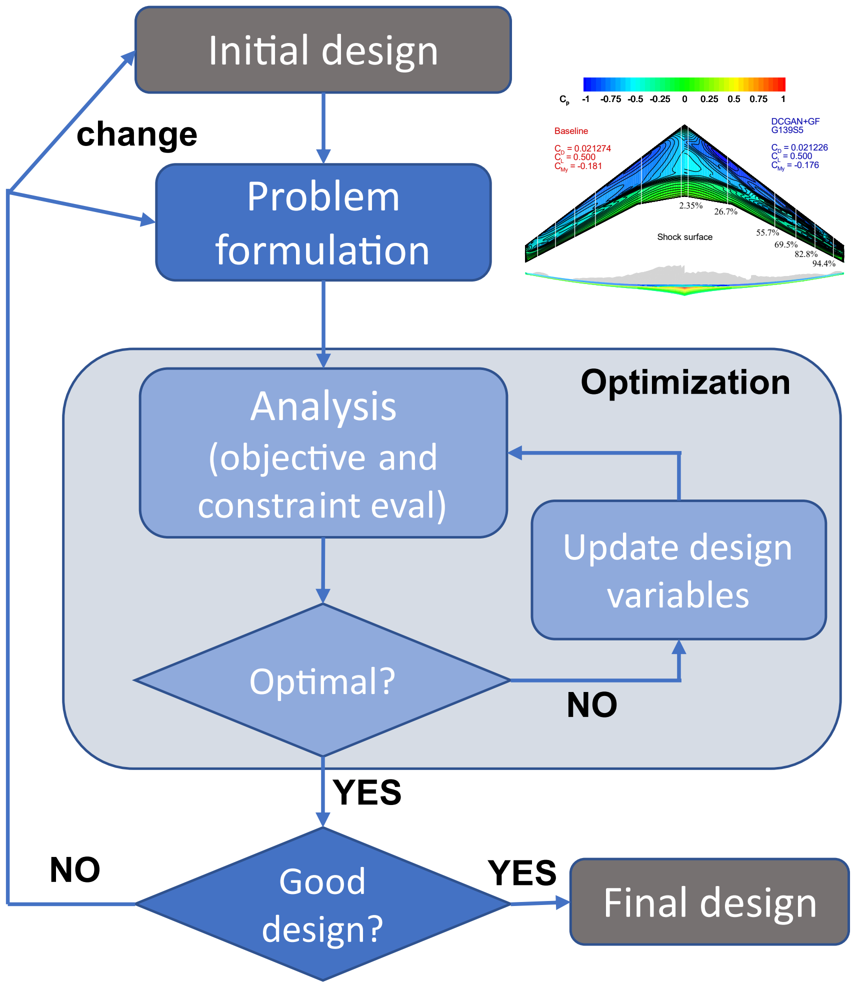
explore many wing shape configs and aerodynamically find the **best one**

High-fidelity CFD for fluid analysis is **expensive** *(hours/days for one design)*

Need for:

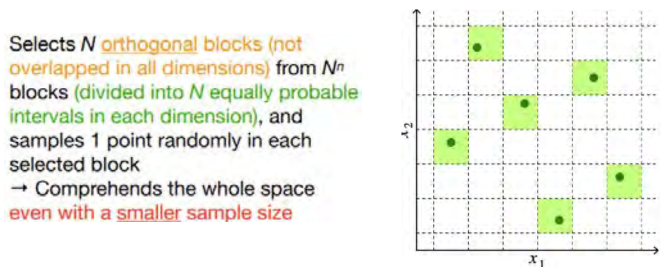
- a fast and accurate fluid analysis
- a more efficient design cycle

Deep Learning

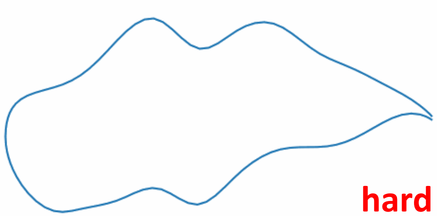


Problems with the conventional method

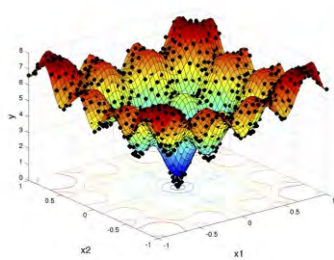
- The conventional way has been using sampling methods like **Latin hypercube sampling (LHS)** in the Design of Experiment + local perturbation.



- **The problems we try to solve:**
  - Expensive fluid analysis
  - Curvy initial designs
  - Inaccurate surrogate models



hard to model



LHS sampling
via FFD local perturbation

In high-dimensional problems
(high flexibility)

Curvy and abnormal initial designs by the LHS.

hard to model

Inaccurate surrogate models.

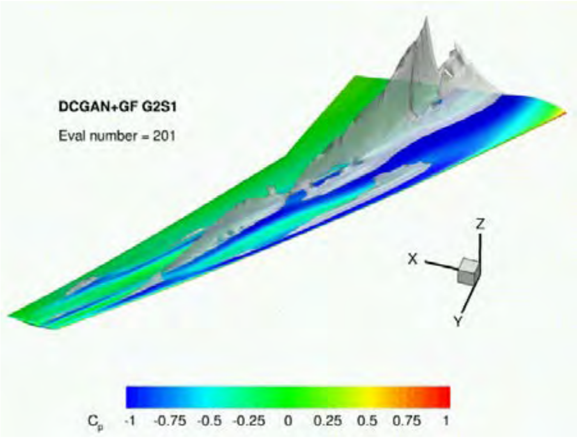
due to the **inaccuracy**

Abnormal infill designs that don't add meaningful information to the models.

Vicious Cycle

Objectives

To improve the efficiency *of the conventional surrogate-based optimization (SBO) method by reducing the required number of CFD analysis while obtaining more optimal design (faster design cycle).*

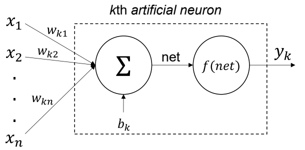


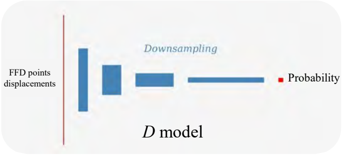
To develop:

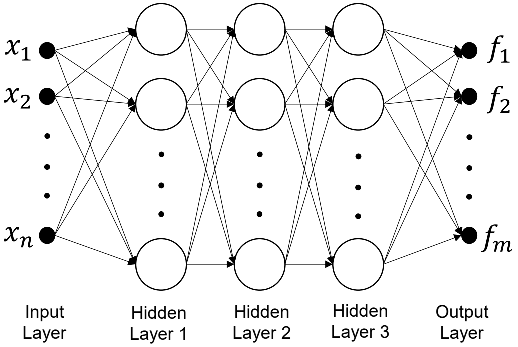
- **M**ulti**l**ayer **p**erceptron
- **D**eep **C**onvolutional **G**enerative **A**dversarial **N**etwork
- **C**onvolutional **N**eural **N**etwork

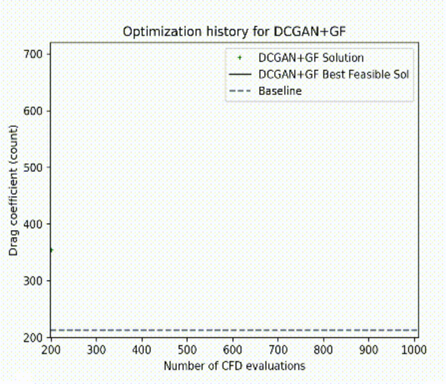
Applied to:

- High-dimensional Aerodynamic Shape Optimization of the Common Research Model Wing





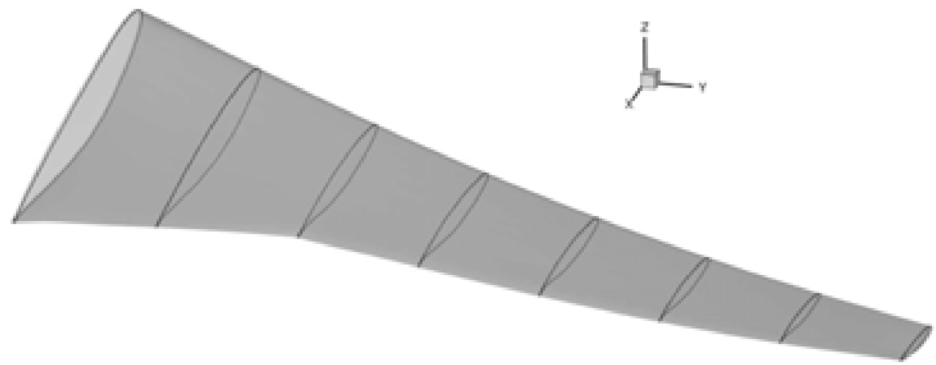


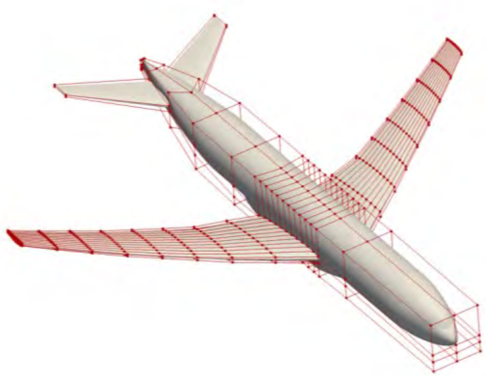


Contents

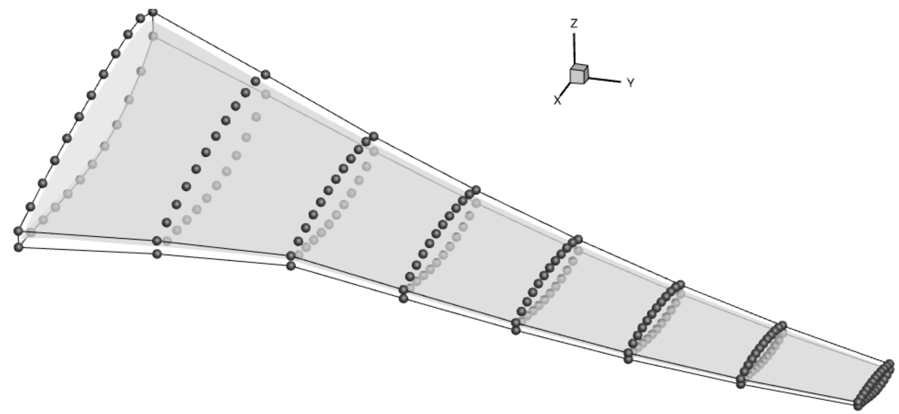
Baseline geometry and parameterization

- A **Common Research Model (CRM)**[1] wing-alone is used as the baseline.
- **Free Form Deformation (FFD)** implemented in pyGeo[2] is used for parameterization.
- By perturbing the FFD points in *z*-direction, a new wing geometry can be obtained.





FFD in pyGeo



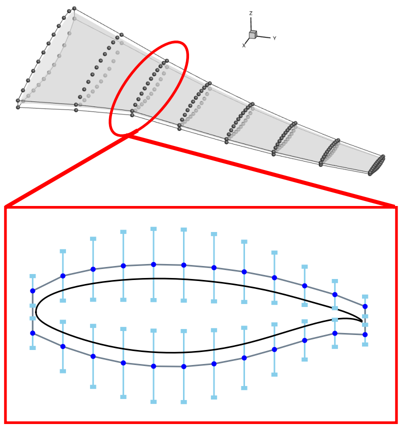
8 sections x **24** points = **192** FFD points embedding the baseline.

1. Vassberg et al., *26th AIAA Applied Aerodynamics Conference*, 2008.
2. https://github.com/mdolab/pygeo

Optimization problem formulation

- We want to **minimize aerodynamic drag**
  - Changing AoA and FFD control points
  - Meeting constraints (lift, moment, geometries)

	Function/variable	Description	Quantity
Minimize	C_D	Drag coefficient	1
With respect to	α	Angle of attack	1
	Δz	FFD control point displacements	192
		Total design variables	193
Subject to	$C_L = 0.5$	Lift coefficient constraint	1
	$C_{My} \geq C_{My,base}$	Moment coefficient constraint	1
	$V \geq 0.8V_{base}$	Minimum FFD volume constraint	1
	$\Delta z_{TE,upper} = -\Delta z_{TE,lower}$	Fixed trailing edge constraints	8
	$\Delta z_{LE,upper} = -\Delta z_{LE,lower}$	Fixed leading edge constraint	1
		Total constraints	12



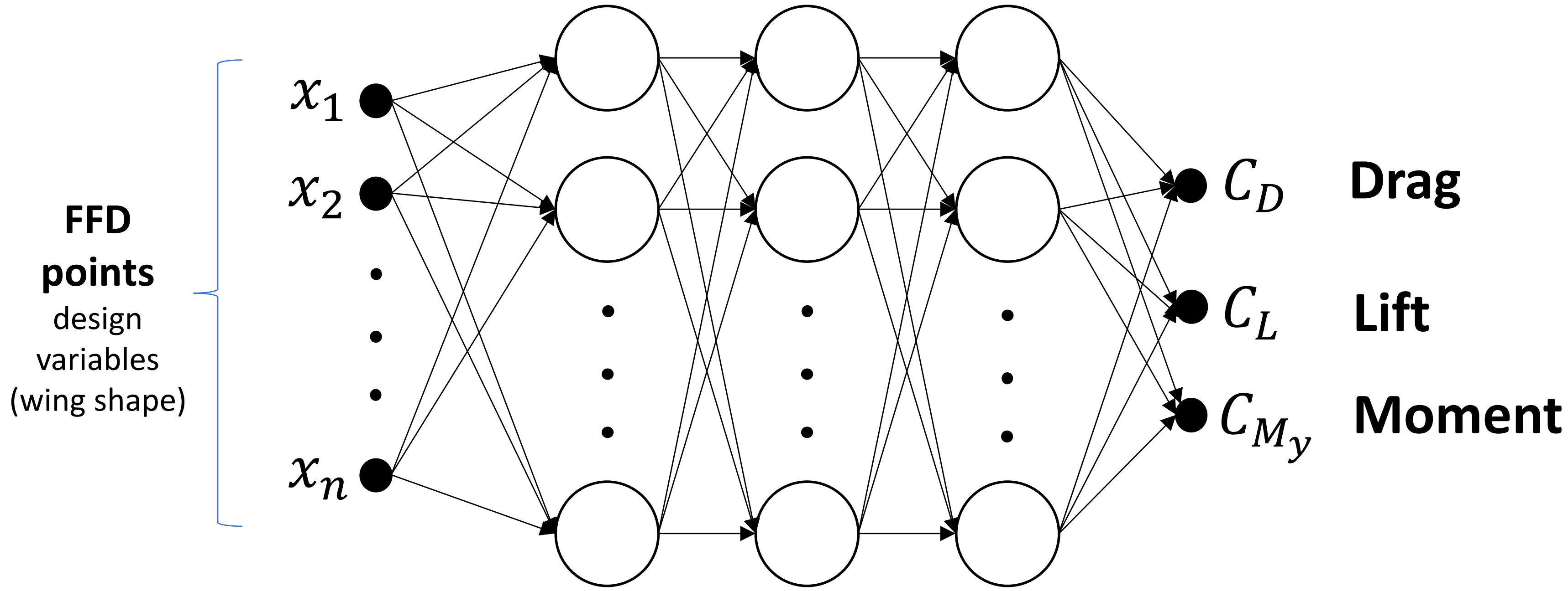
FFD boundaries of the 3rd spanwise section

> 100 design variables high-dimensional!

Contents

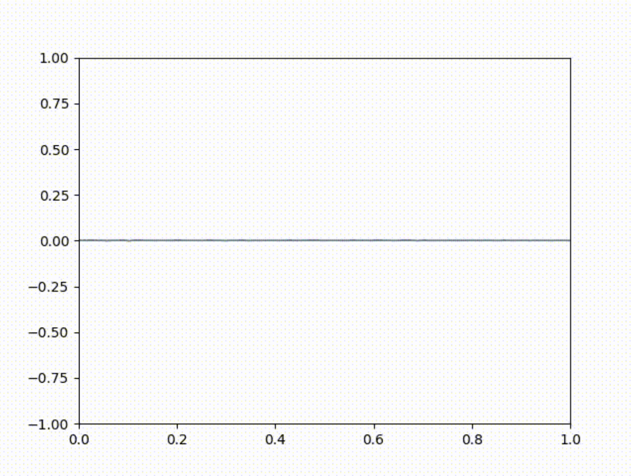
MLP-based surrogate modeling

- **MLP**: Multilayer Perceptron
- Computational Fluid Dynamics → **expensive**
- MLP provides fast analytical prediction as a surrogate, instead of CFD.

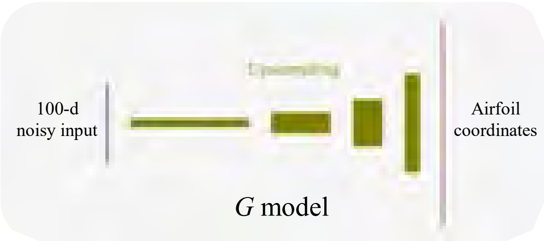


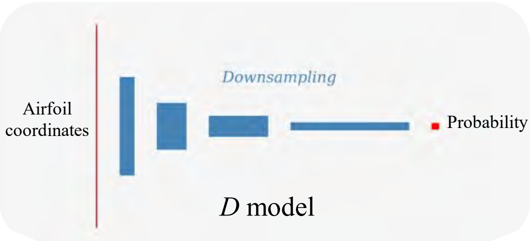
DCGAN-based sampling (1/2)

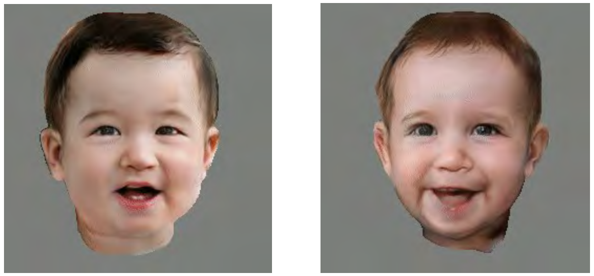
- **Deep Convolutional GAN (DCGAN)**: a type of GAN[3] that consists of only convolutional layers.
- We trained the DCGAN using the 77 transonic airfoils from the **UIUC database**.
- After training, the *G* model can produce **synthetic airfoil** coordinates from 100-dimensional noise.



Airfoil transformation from a noisy input by our GAN





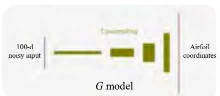


Fake images produced by a GAN[4]
Lifespan Age Transformation Synthesis

3. Goodfellow et al., *Advances in Neural Information Processing Systems*, 2014.
4. Or-El et al., https://arxiv.org/pdf/2003.09764.pdf

DCGAN-based sampling (2/2)

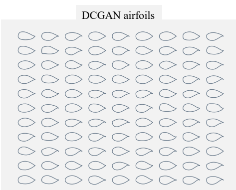
1

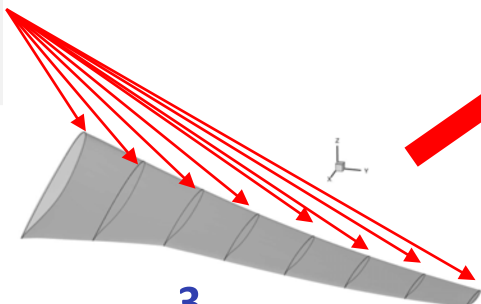


$$A_{(mxn)} = \begin{bmatrix} a11 & a12 & . & . & . & a1n \\ a21 & a22 & . & . & . & a2n \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ am1 & am2 & . & . & . & amn \end{bmatrix}$$

8 x 100 random noises

2





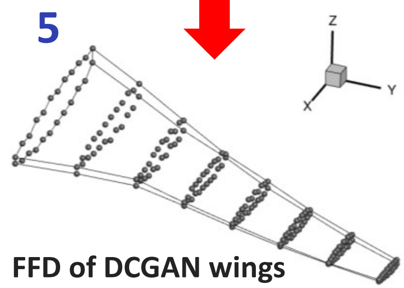
3

4

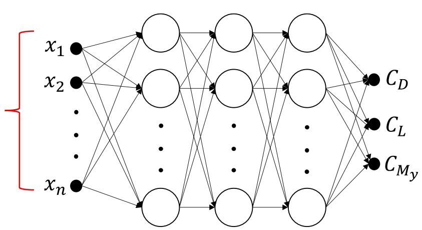
Algorithm 1 – The search of the corresponding FFD control points given a set of 8 DCGAN airfoils.

procedure INVERSE_FFD(z_t) ⇨ z_t is the target DCGAN airfoil z-coordinates
  for $i = 1; i \leq 8; i++$ **do** ⇨ loop over the 8 airfoils
    $z_t = \text{scale}(z_t)$ ⇨ scale the airfoils to have the size like baseline
    $z_t = \text{reposition}(z_t)$ ⇨ translate+rotate so the TE position is like baseline
  end for
  start optimization ⇨ perform an optimization using NSGA-II
    minimize $|z - z\_t|$ ⇨ z is the candidate coordinates after perturbing FFD
    with respect to FFD ⇨ FFD is the candidate FFD control point coordinates
    subject to geometric constraint ⇨ the volume constraint
  end optimization
  return FFD ⇨ return the FFD control point coordinates
end procedure

5



FFD of DCGAN wings

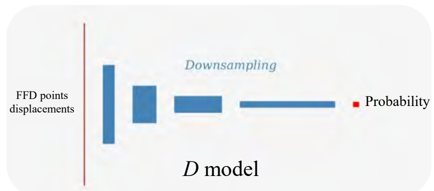


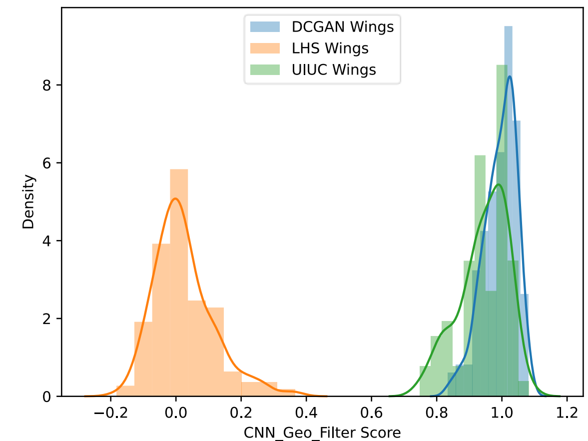
Surrogate Model

DCGAN-based sampling
1) Produce 8 x 100 random noises
2) Produce 8 airfoils using *G* model
3) Wing representation
4) Perform Algorithm 1
5) Obtain the FFD points

CNN-based geometric filtering

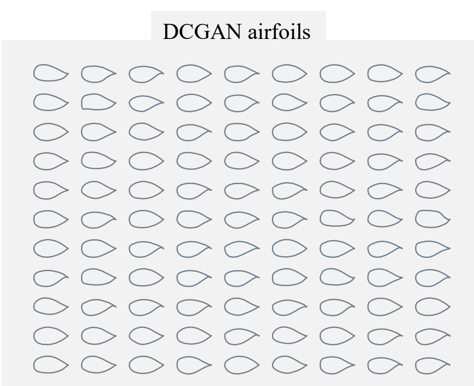
- **CNN: Convolutional Neural Network**
- We trained it using 500 smooth and curvy samples.
- Smooth score label → **1**
- Curvy score label → **0**



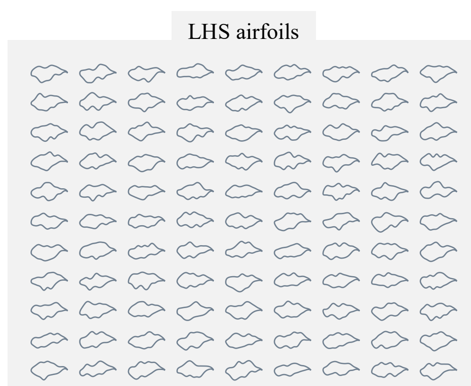


Smooth → scores near one
Curvy → scores near zero

- After training, the CNN can detect the shape abnormality by giving a probability score.



realistic



abnormal

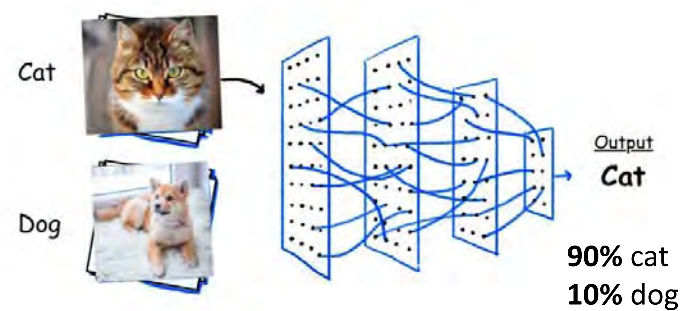
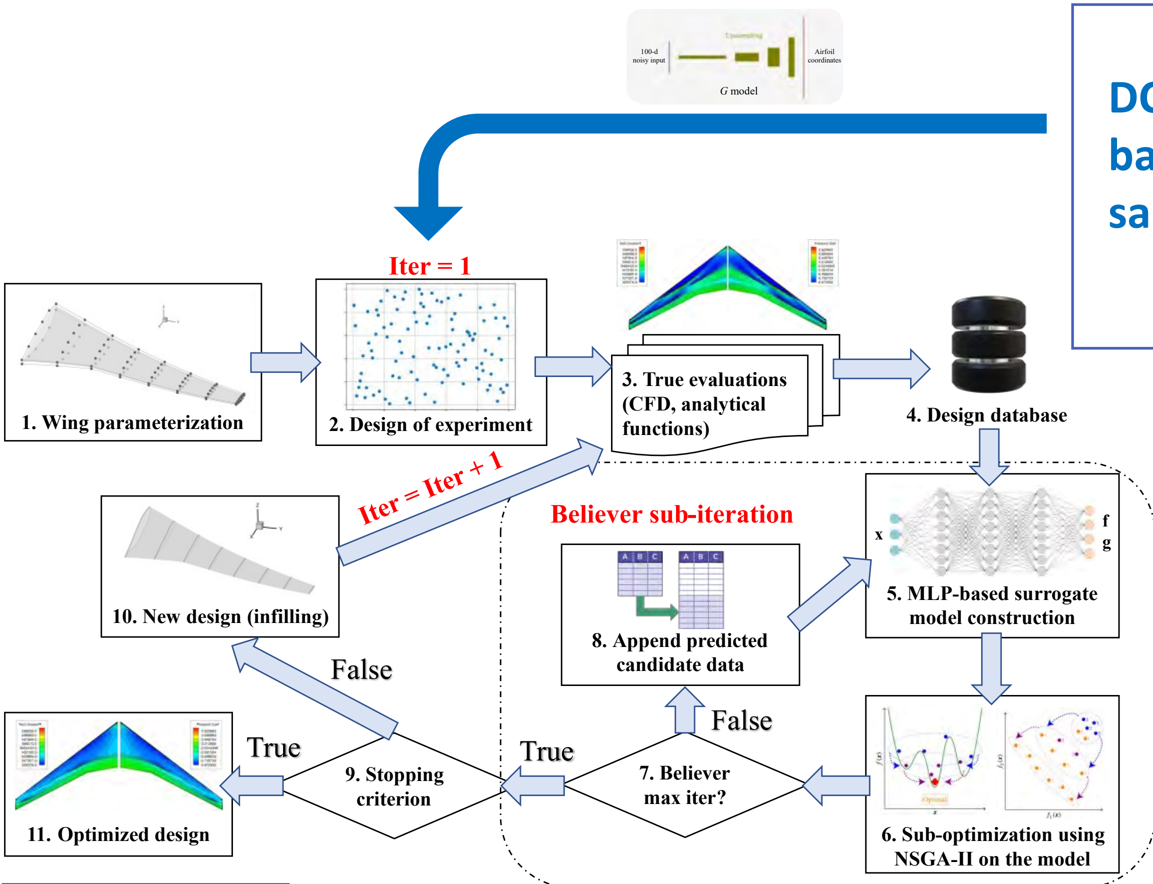
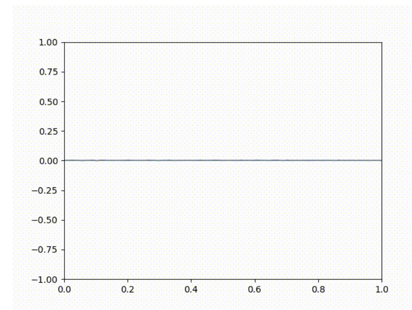


Image classification by CNN
Credit: Yanjia Li, towardsdatascience

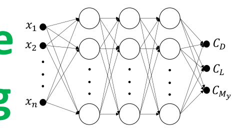
The entire design framework



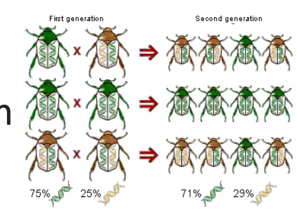
DCGAN-based sampling



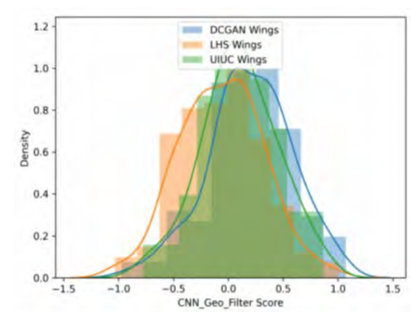
Surrogate modeling

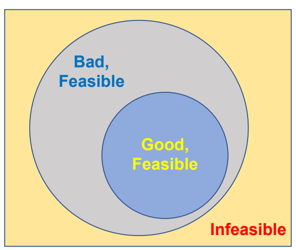


Genetic algorithm
NSGA-II



CNN-based filter



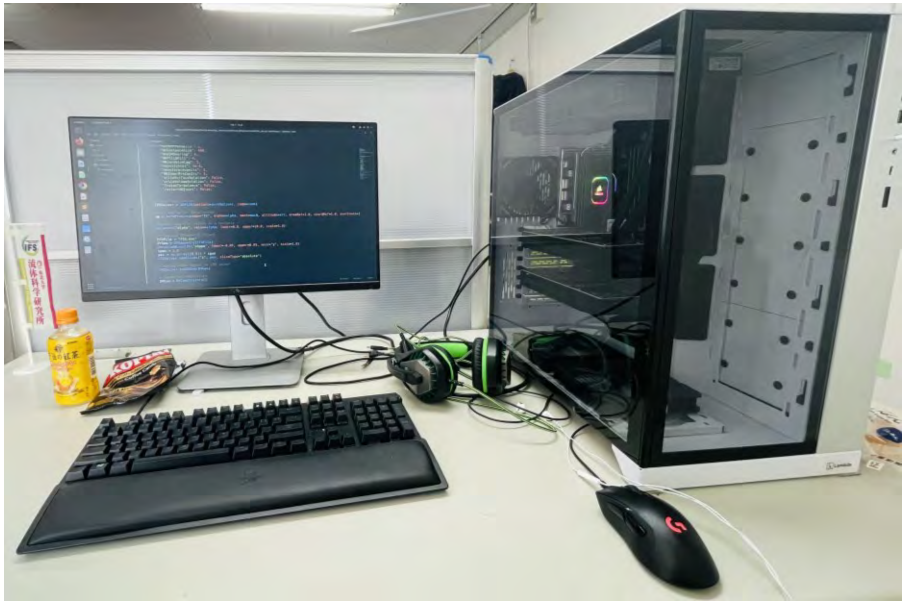


a value that reduces the design space for a more accurate modeling

Analytical constraint

$$S \geq S_{cut-off} = 0.4$$

Computational resources



- CFD analyses are parallelly performed in an AFI System[5] of the *Institute of Fluid Science, Tohoku University.* Each analysis took 30 minutes on an Intel Xeon Gold 6148 2.4GHz with 40 processors.

- Deep learning models are trained on a local computer with 2xGPU RTX3070. MLP training took an order of seconds, while CNN and DCGAN training took an order of minutes to hours.

5. http://www.ifs.tohoku.ac.jp/~afirc/afirc_eng/supercomputer/

Contents

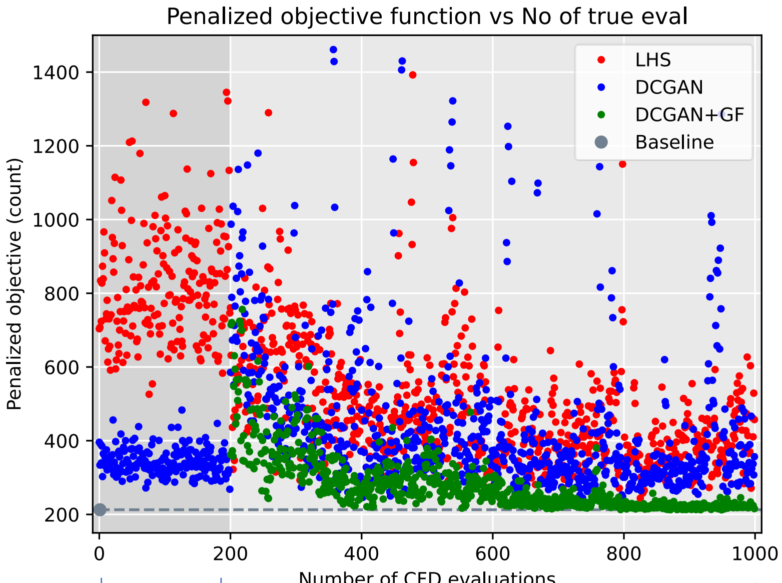
Optimization history and surrogate model's accuracy

- **Experiments**:
  1. **LHS:** using Latin hypercube sampling
  2. **DCGAN:** using the DCGAN technique
  3. **DCGAN+GF**: using the DCGAN+CNN technique

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

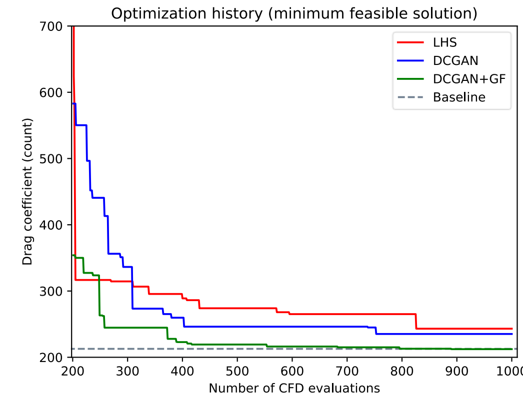
$$Penalized\ objective = 10{,}000 \times C_D + 1000 \times |0.5 - C_L| + 1000 \times \left|\min\left(C_M - C_{M,base}, 0.0\right)\right|$$

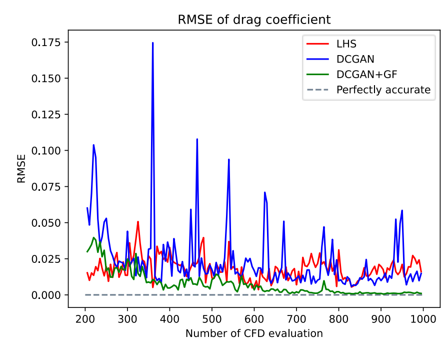
- The DCGAN+CNN improves the optimization performance.

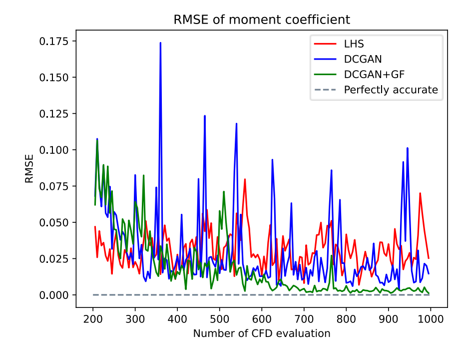


initial samples | **infill samples**

LHS: conventional
DCGAN: proposed 1
DCGAN+GF: proposed 2



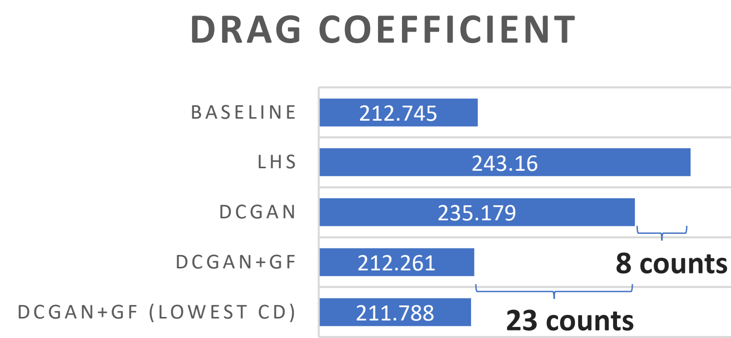




CNN geo filter → **speeds up convergence and improves the model's accuracy**

Optimal solutions comparison

- The **DCGAN-only** → **8 counts** less drag than the **LHS**.
- The **DCGAN+GF** → **23 counts** less drag the DCGAN-only.
- The lowest drag solution → **1 count** less drag than the baseline but **violating** the moment constraint.
- The best feasible → **0.5 count** less drag than baseline.



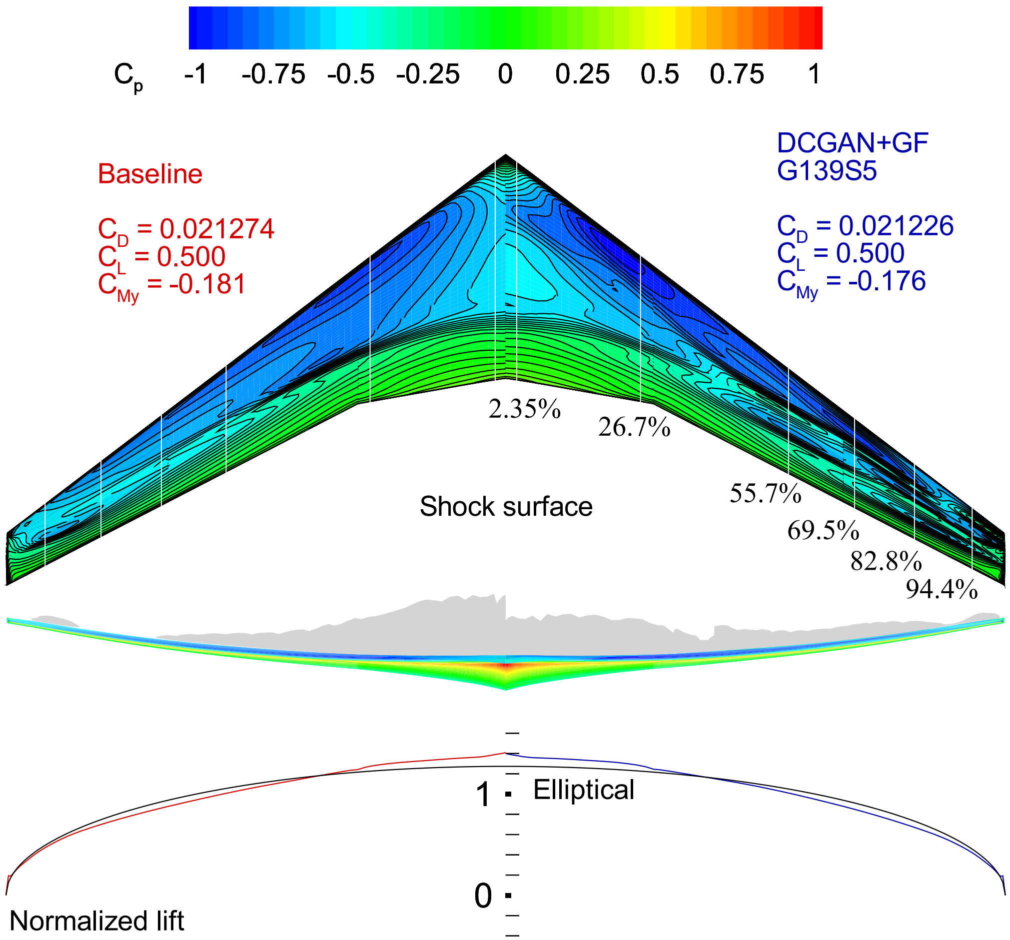
		Feasible		best	Infeasible
	Baseline	LHS	DCGAN	DCGAN+GF	DCGAN+GF (lowest drag)
Design	-	G127S1	G112S3	G139S5	G136S2
C_D (counts)	212.745	243.160	235.179	212.261	211.788
C_L	0.500	0.499	0.500	0.500	0.500
C_{My}	-0.181	-0.168	-0.174	-0.176	-0.182
AoA	2.212°	2.286°	2.267°	2.207°	2.217°
FFD Volume	0.689	0.722	0.709	0.700	0.687
Wing Volume	0.231	0.232	0.232	0.230	0.230

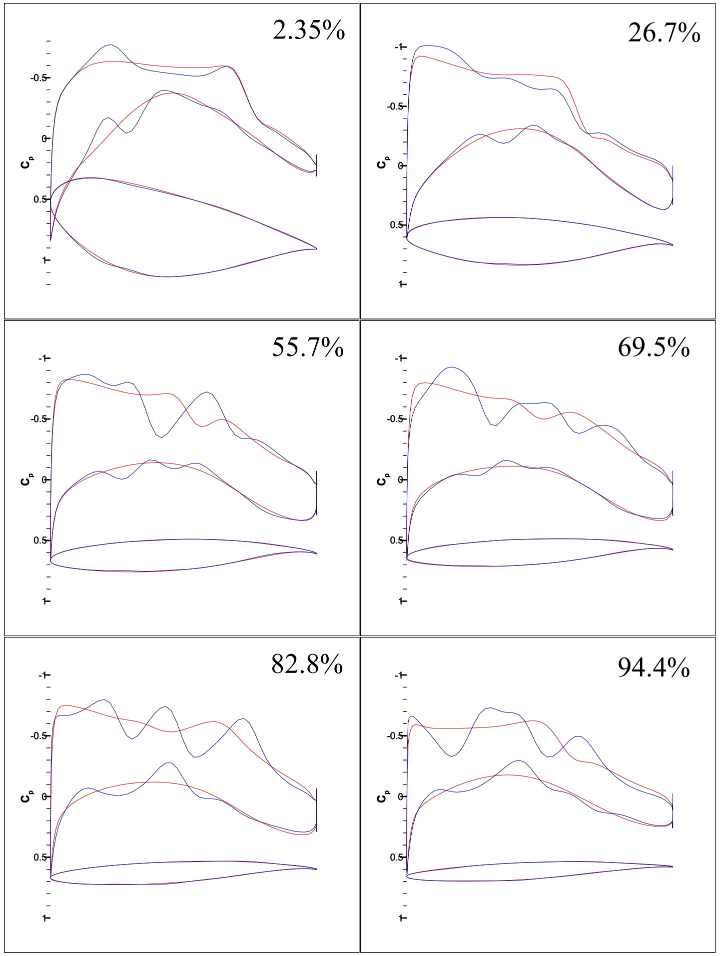
conventional **proposed**

CFD: Baseline vs DCGAN+GF best feasible (1/2)

DCGAN+GF best feasible has

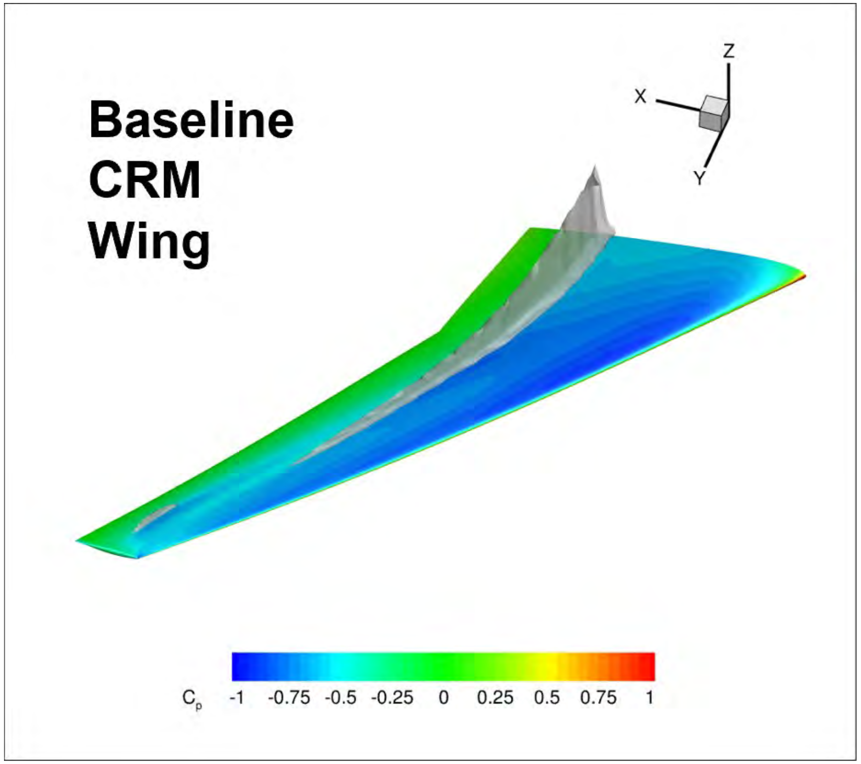
- **0.5 count** less drag than baseline
- **Less intense** shock at 0 – 45% spanwise location

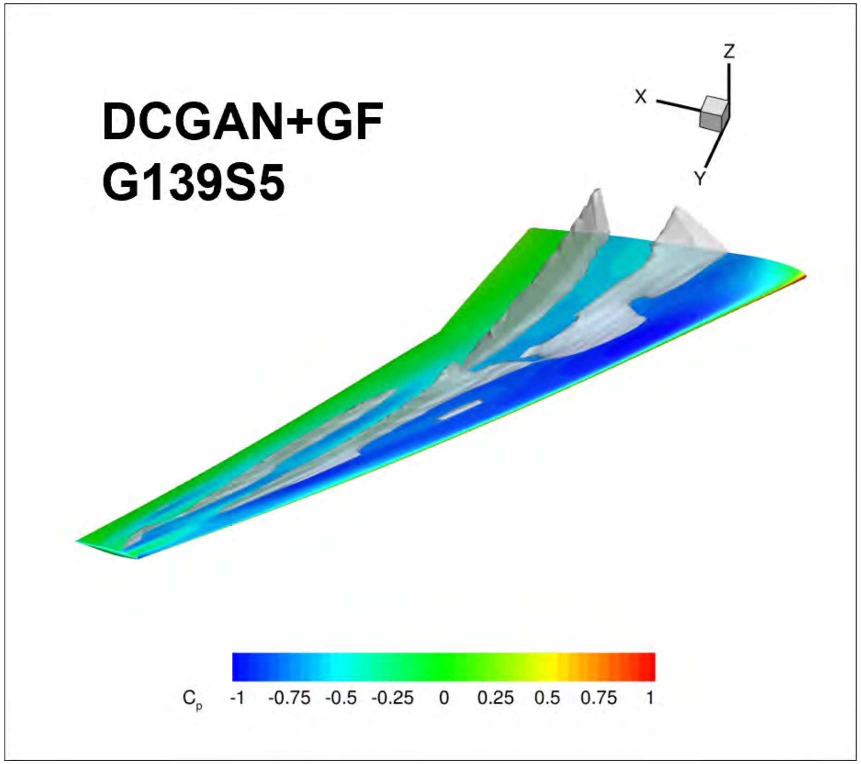




CFD: Baseline vs DCGAN+GF best feasible (2/2)

DCGAN+GF optimal design exhibits a **bifurcated** shock region, resembling a λ-configuration.



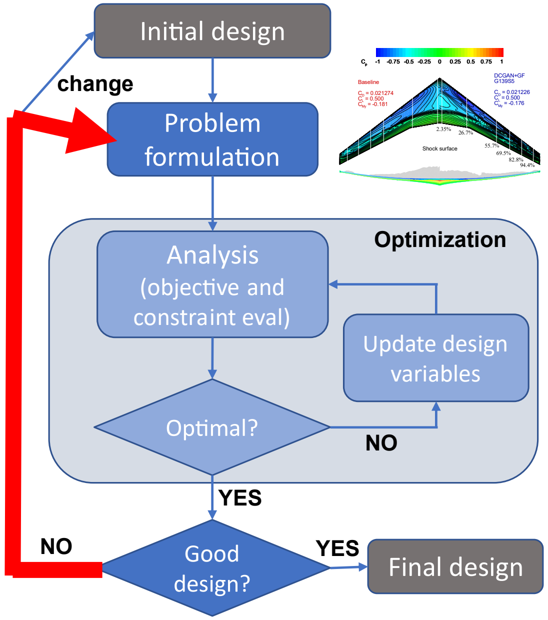


Is a 0.5-count improvement from the baseline worthwhile?

- Comparing our results with other researchers

Researcher	Improvement (drag count)	Number of design vars (FFD points)	CFD samples	Using adjoint	Optimizer
My case [6]	0.5	8 x 24 = **192**	1000	No	Gradient-free
Li et al. [7]	0.7 – 2.0	8 x 24 = **192**	1000	Mixed	Gradient-based
Lyu et al. [8]	16.0	15 x 48 = **720**	800	Yes	Gradient-based

- To get a better result, we should **revise the problem formulation** (as part of the design cycle): increase the FFD points, etc.
- CRM wing is **already a good-performing** design to start with, developed by experts.
- Our project focuses on **proving the efficacy of the deep learning techniques** in improving the efficiency of *gradient-free adjoint-free surrogate-based optimization (SBO) methods with **LHS**.*



6. Hariansyah et al., *Proceedings of the 33rd Congress of ICAS*, 2022.
7. Li et al., *AIAA Journal*, Vol. 59, No. 6, 2021.
8. 10. Lyu et al., *AIAA Journal*, Vol. 53, No. 4, pp 968-985, 2015.

Contents

Concluding remarks

PROBLEMS

We are facing:

- CFD analysis is expensive.
- Latin hypercube sampling (LHS) + FFD produces designs with curvy surfaces.
- In high-dimensional case, it's difficult to build an accurate surrogate model.

SOLUTIONS

We introduced:

- MLP-based surrogate modeling
- DCGAN-based sampling
- CNN-based geometric filtering

Used together, the above techniques could **speed up the optimization convergence**.

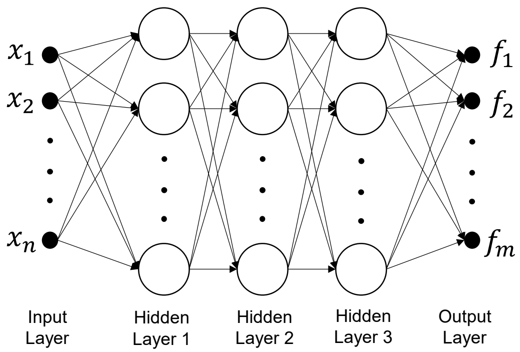
Thank you for your support!

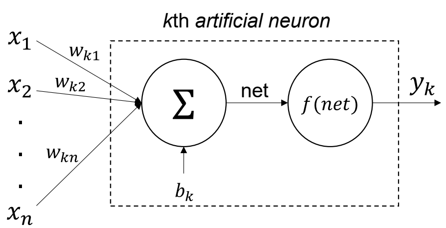
33rd Congress of the International Council of the Aeronautical Sciences, September 2022, Sweden

Deep learning techniques summary

a. Multilayer perceptron

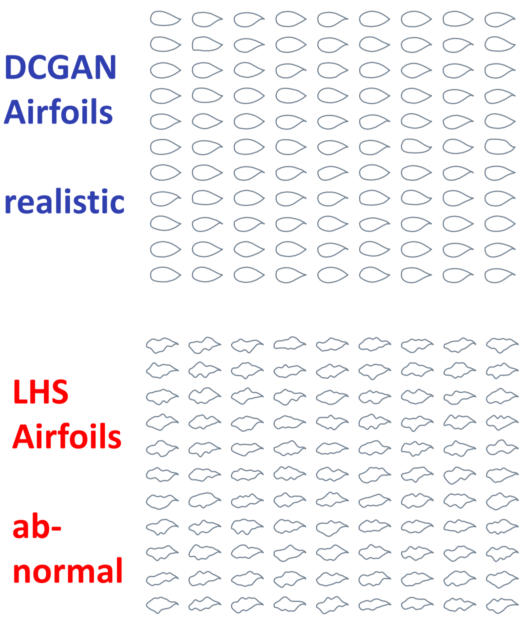
- Replace CFD in the GA process, reducing the turnaround time





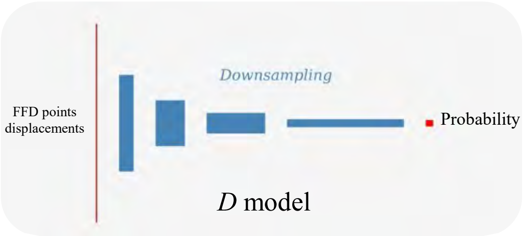
b. DCGAN-based sampling

- Provide good initial designs (smoother wing surfaces and better performance)

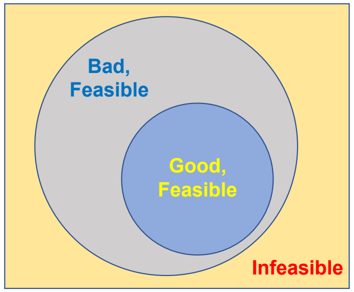


c. CNN-based filtering

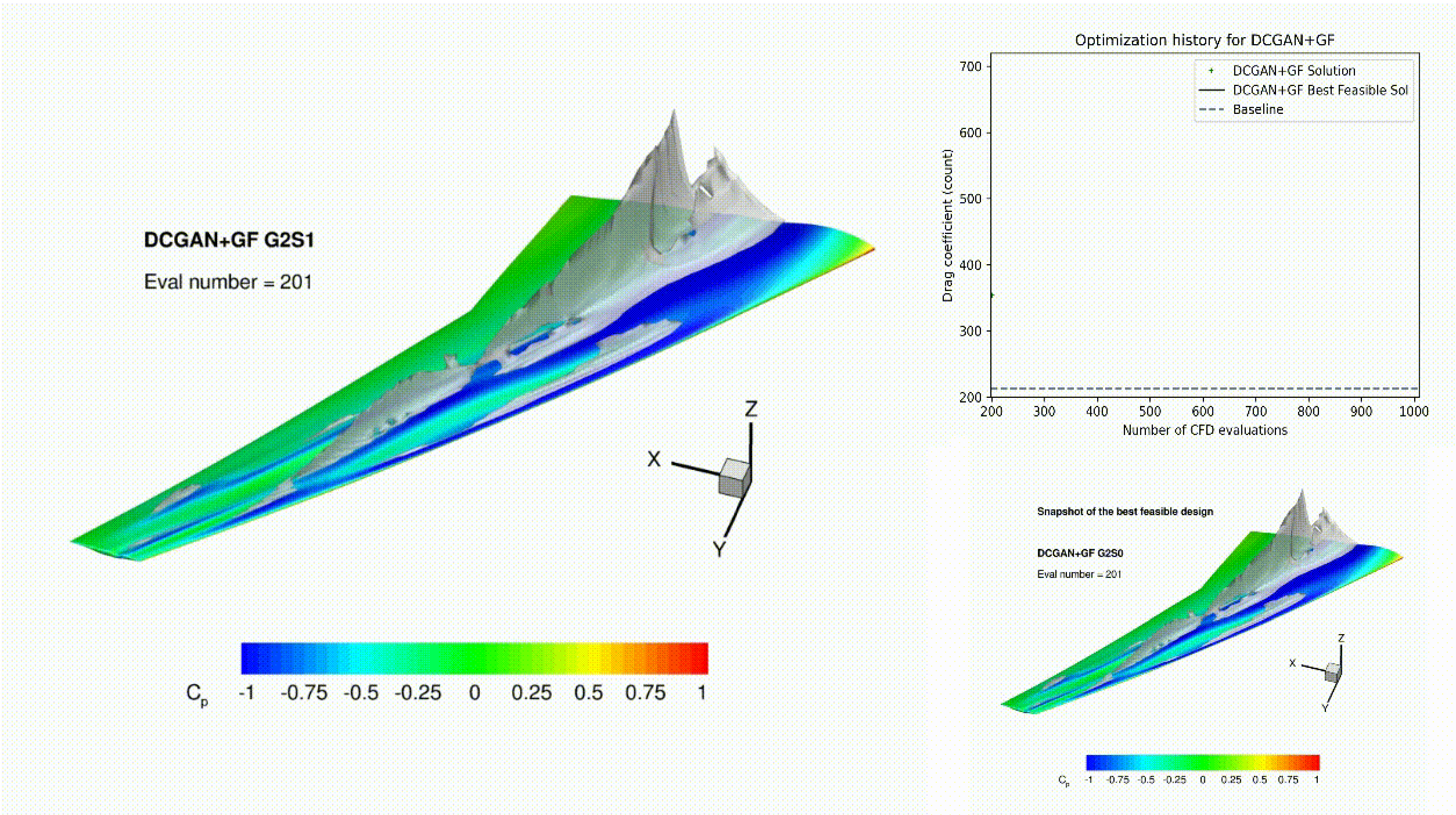
- Shrink the design space, increasing the surrogate model's accuracy



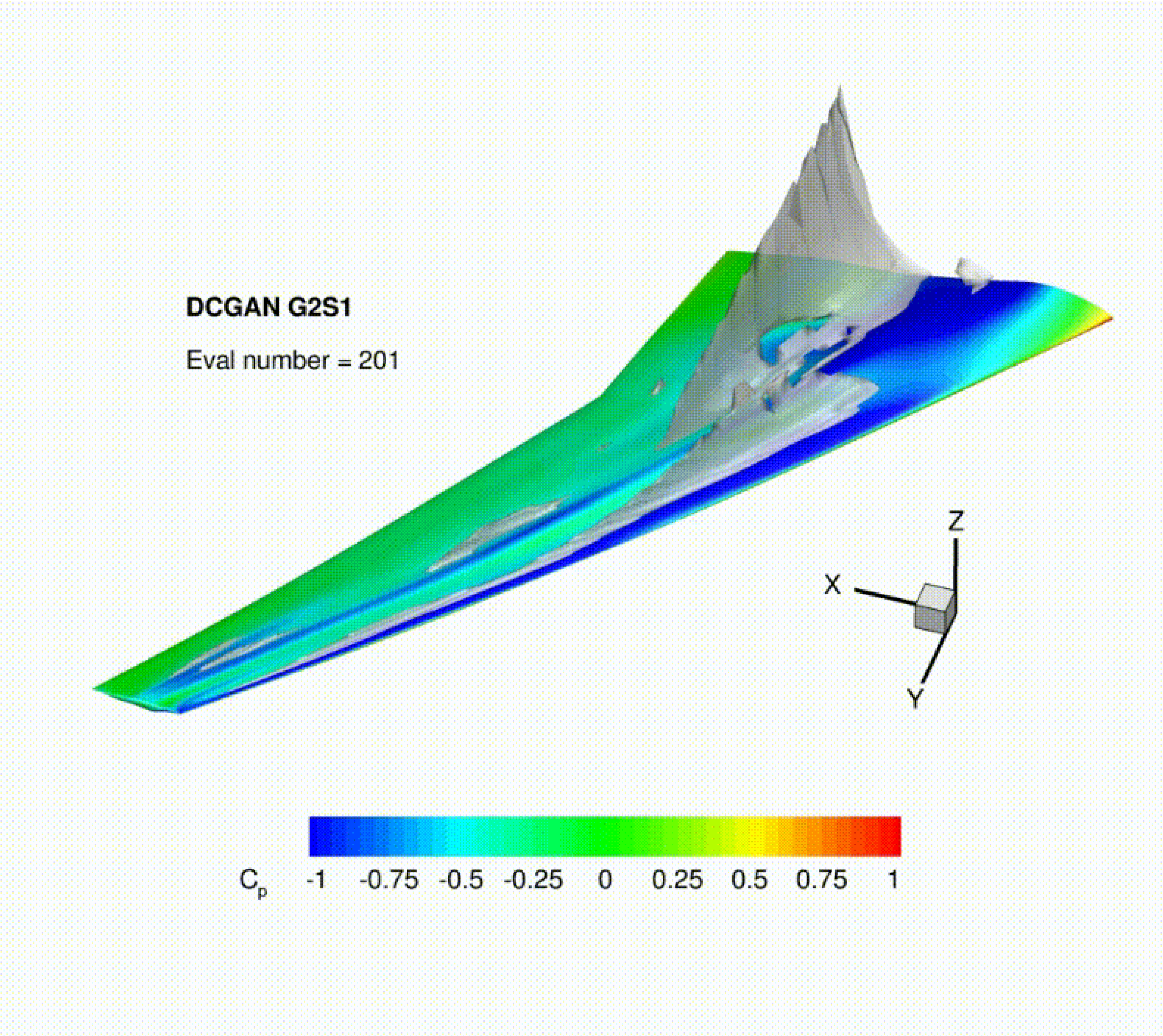
Analytical constraint
$S \geq S_{cut-off}$

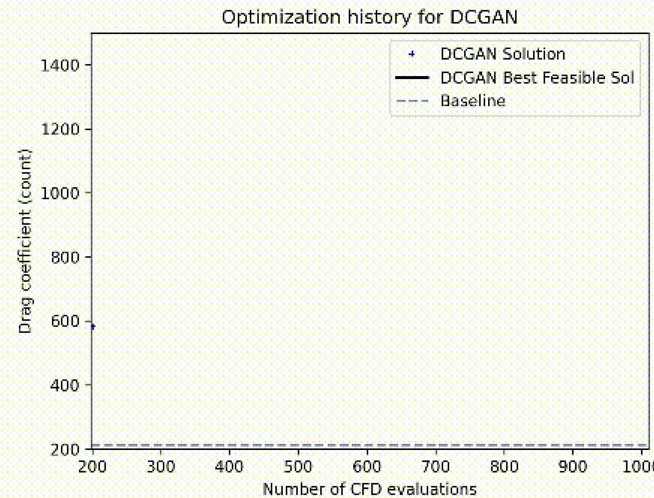


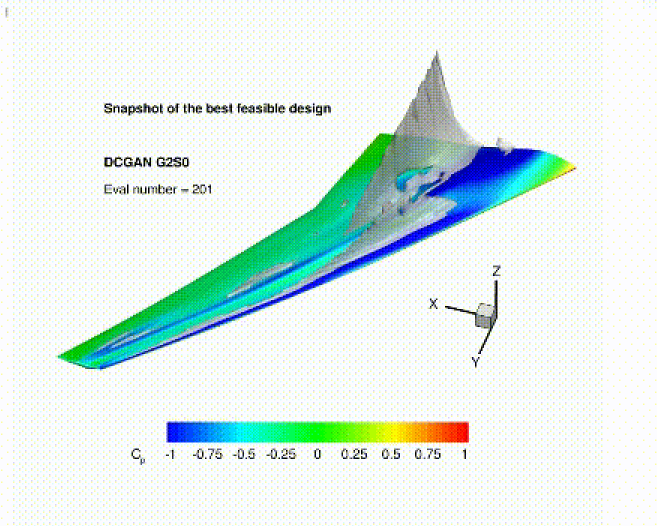
DCGAN+GF Optimization



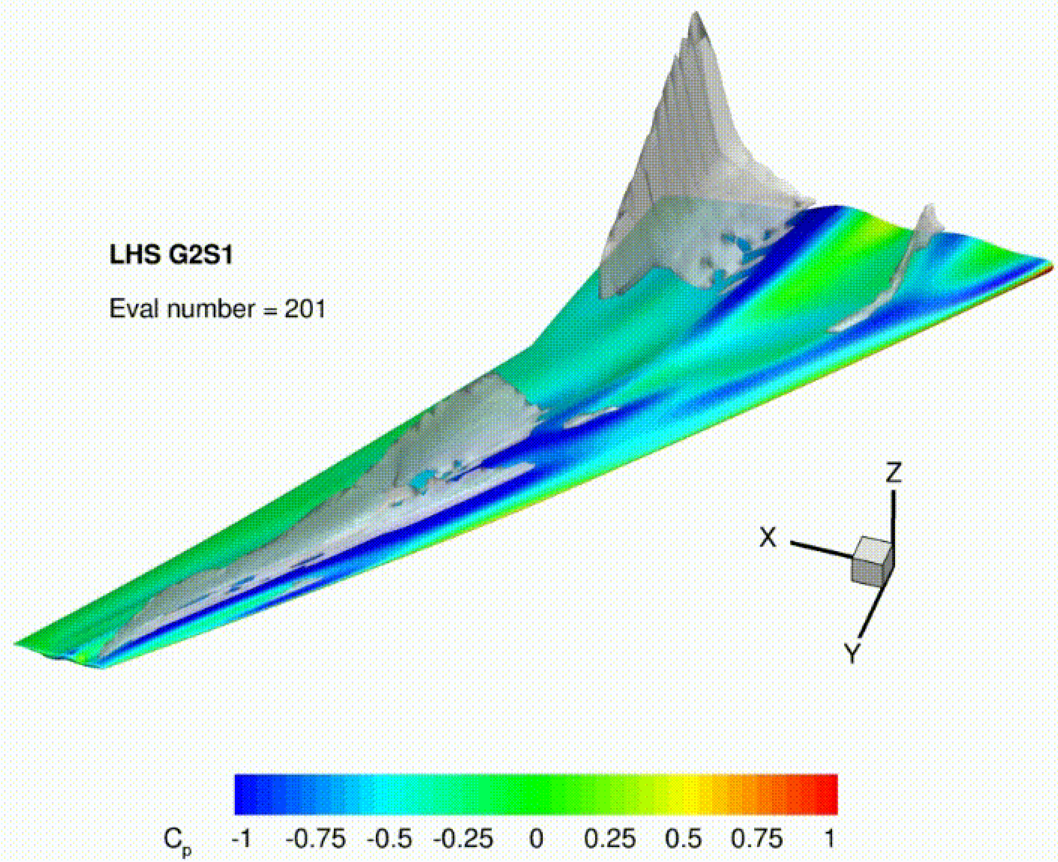
DCGAN Optimization

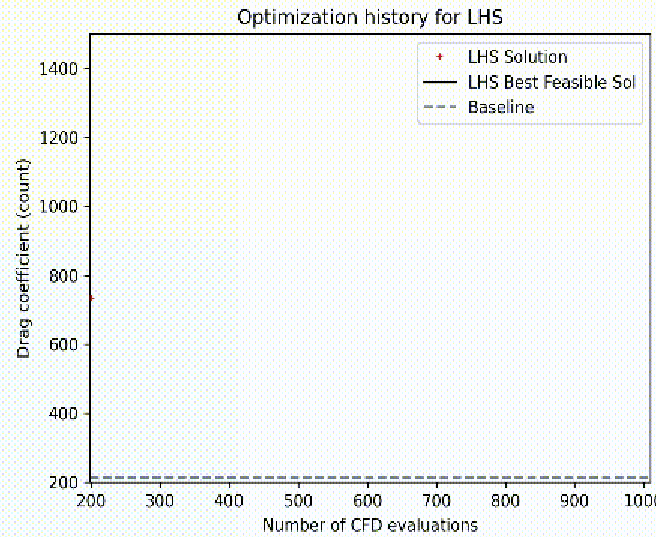


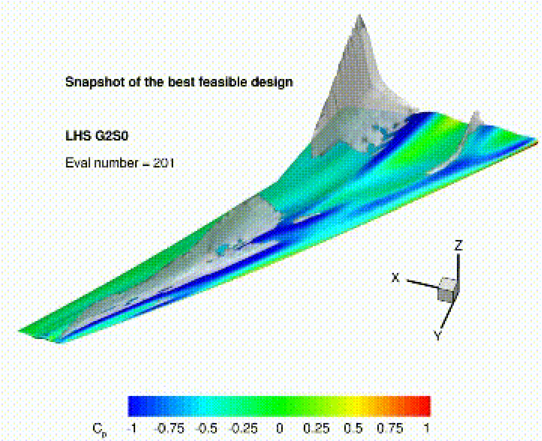




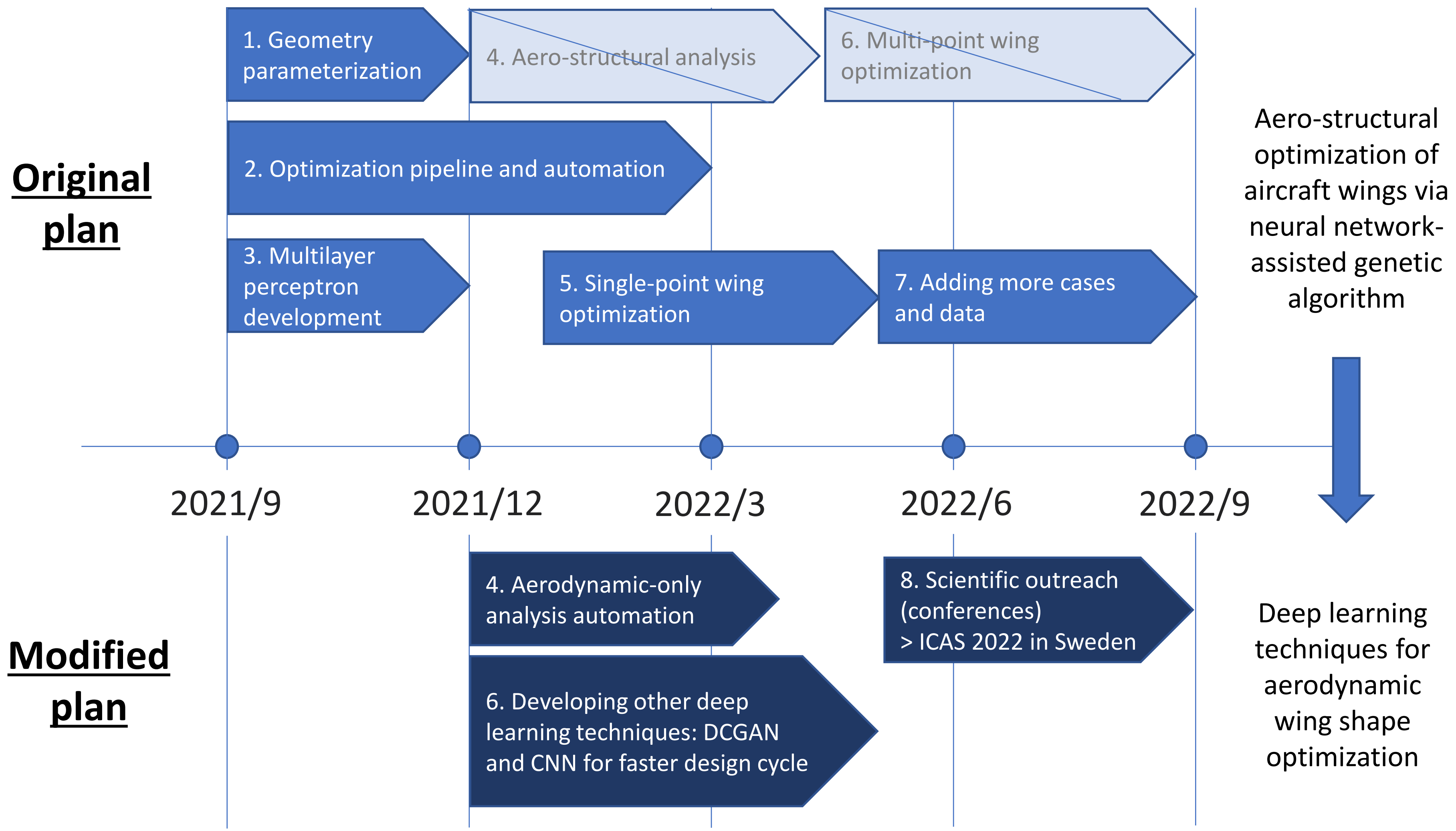
LHS Optimization







Backup slides (1/8): Project plan

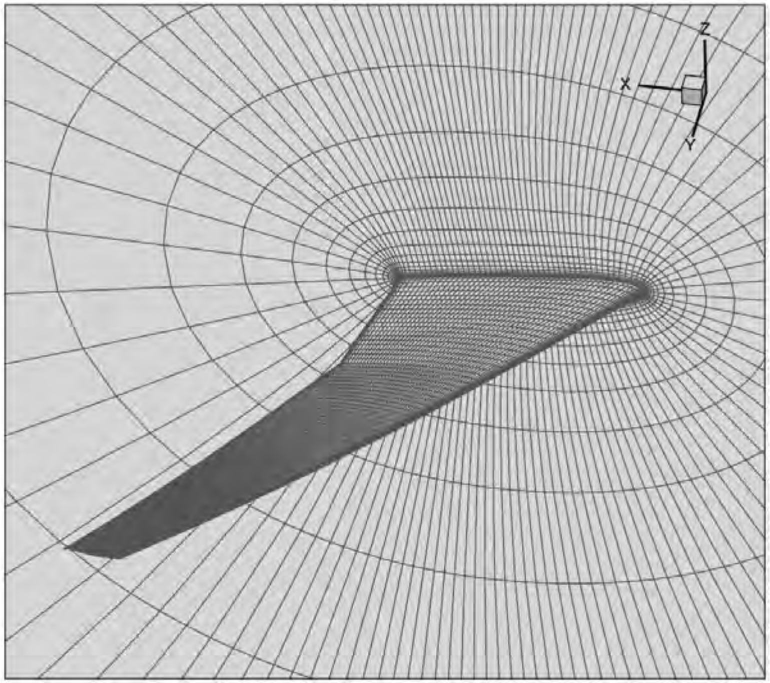


Backup slides (2/8): CFD and mesh deformation

- A finite-volume RANS CFD solver (ADflow[9]) from MDOLab is used.

- Flight conditions:
  - Mach 0.85, Re = 5 x 10^6
  - Fixed C_L = 0.5
  - AoA initial guess = 2^o

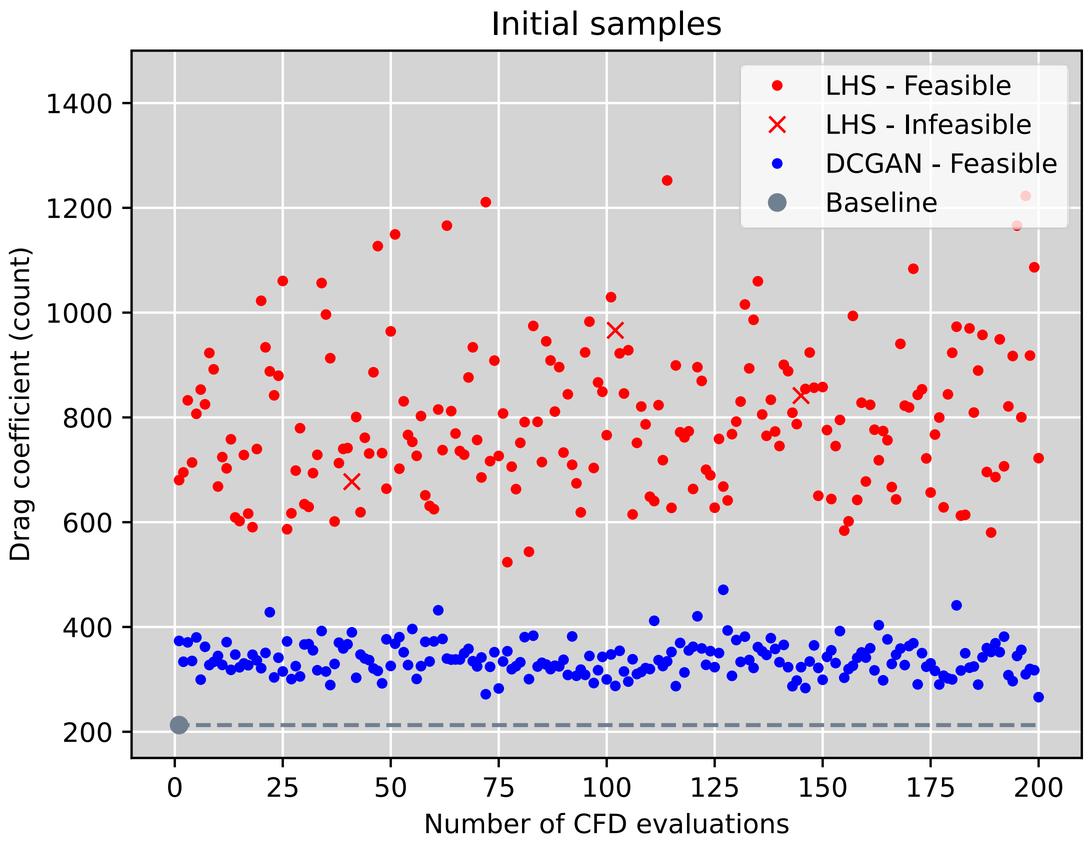
- Given displacements in FFD points, the baseline mesh is deformed using IDWarp[10] to produce a new volume mesh for the new design.



The baseline mesh with approximately 450 thousand cells

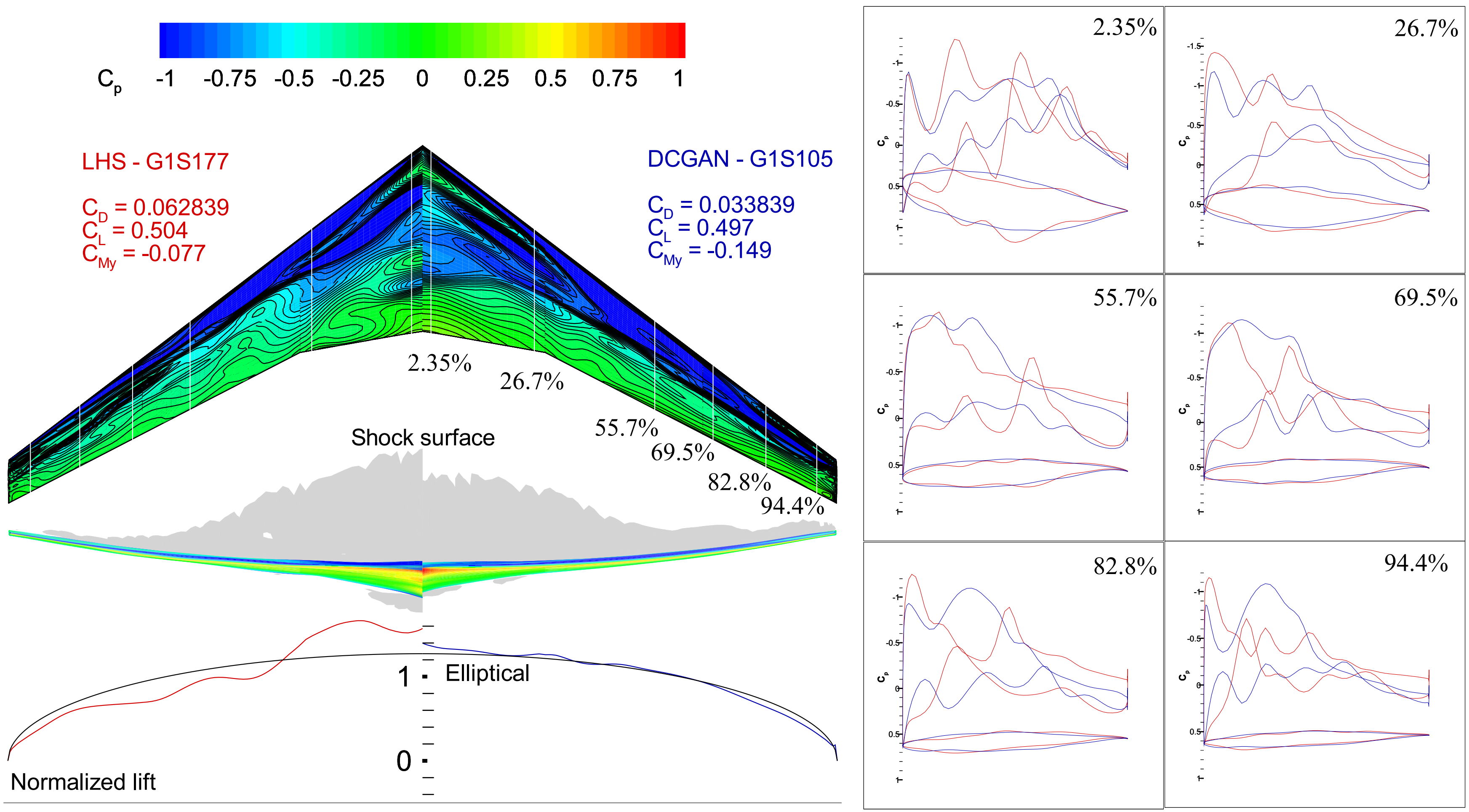
9. https://github.com/mdolab/adflow
10. https://github.com/mdolab/idwarp

Backup slides (3/8): DCGAN vs LHS initial samples distribution

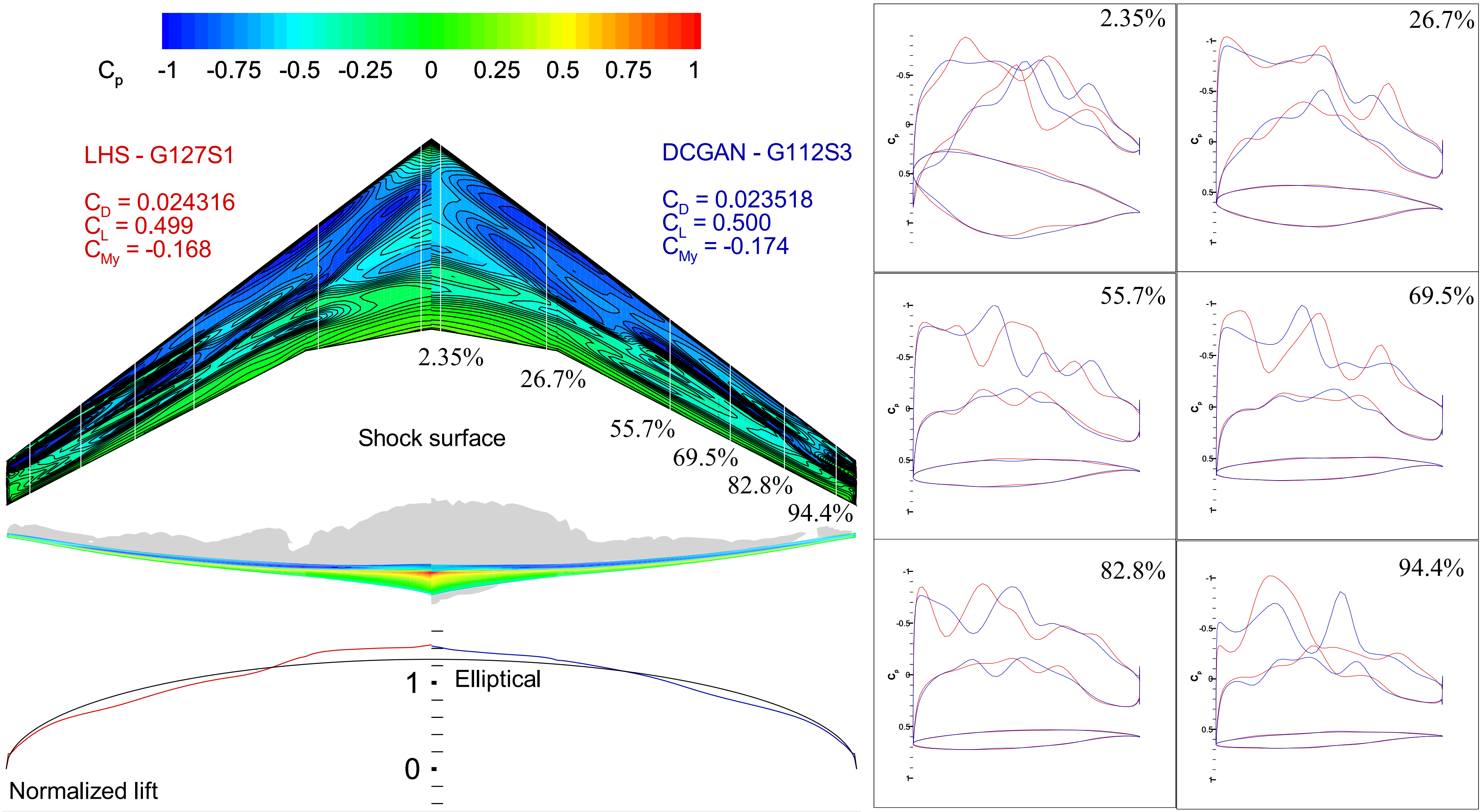


Backup slides (4/8): CFD: LHS vs DCGAN initial samples

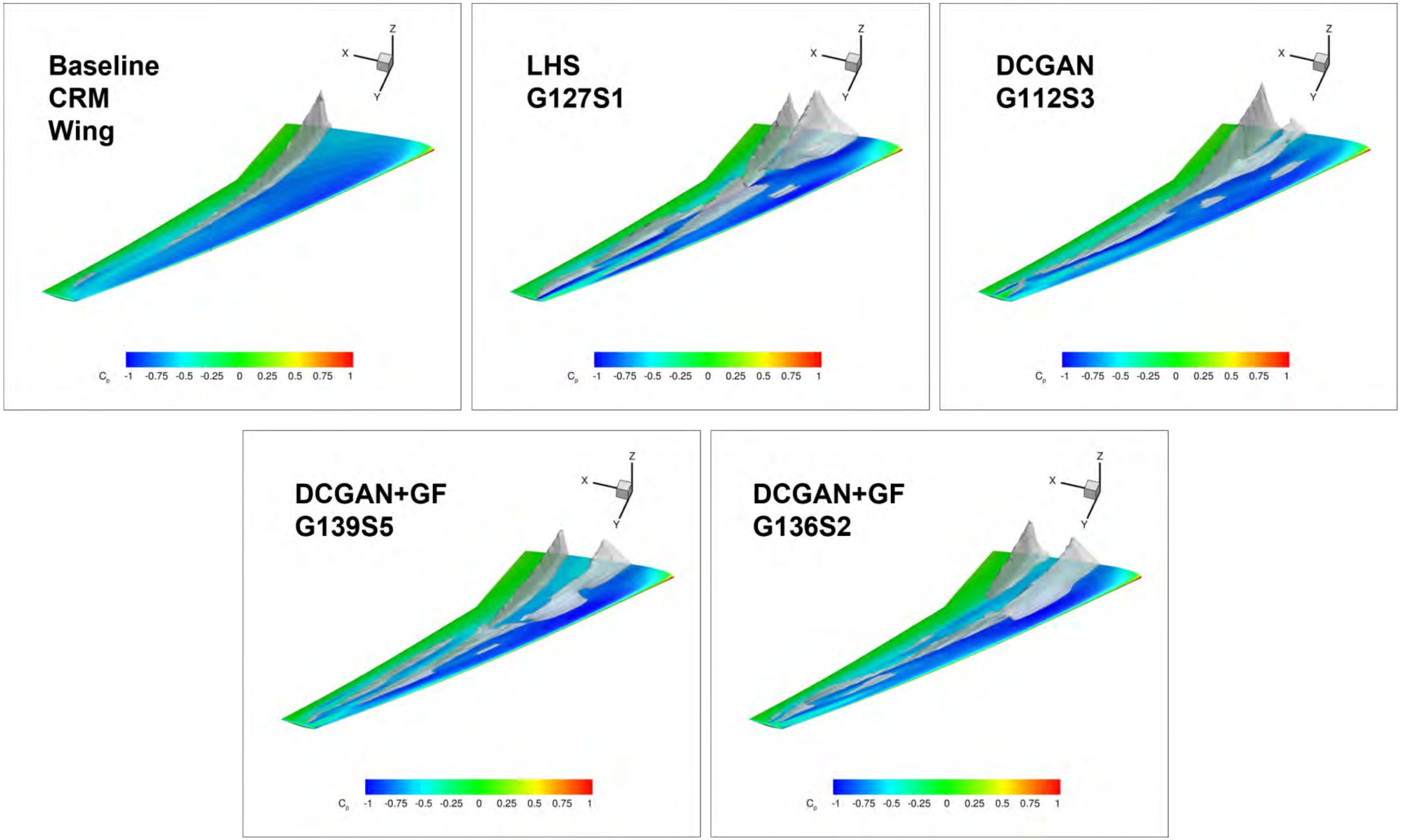
DCGAN initial samples → **smoother** than LHS initial samples → **better** aerodynamic performance



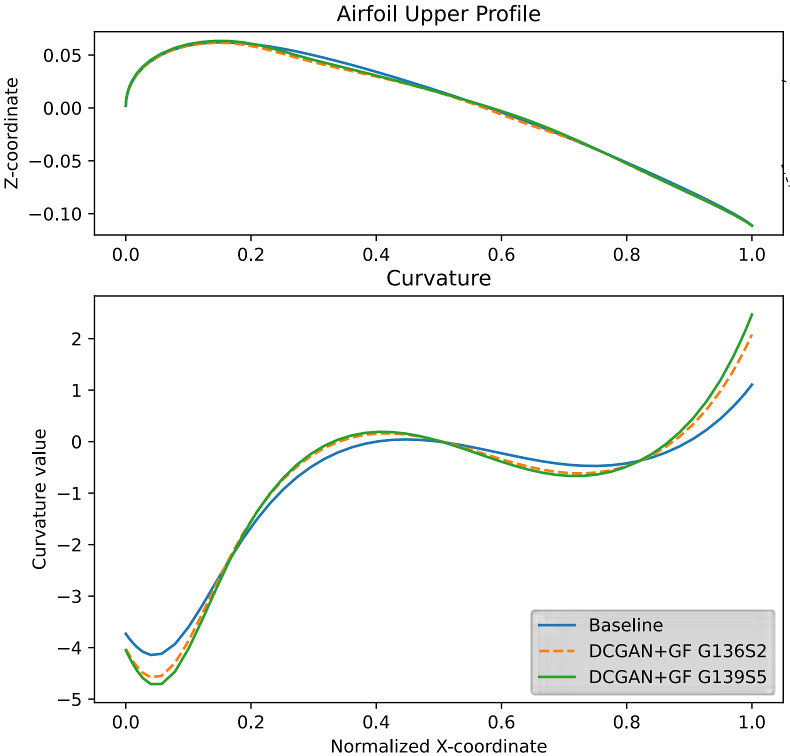
Backup slides (5/8): DCGAN vs LHS optimal solutions

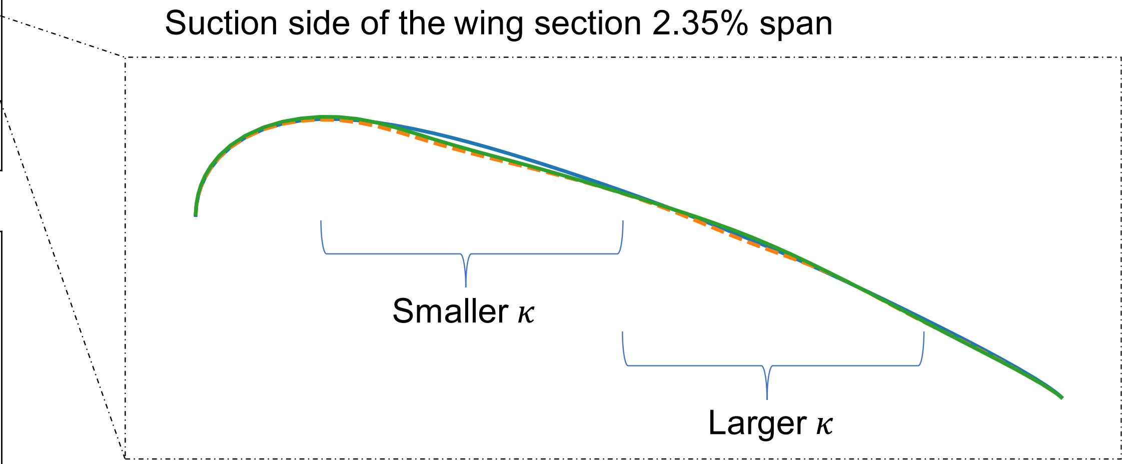


Backup slides (6/8): 3D shock visualization



Backup slides (7/8): curvatures





$$\kappa(x) = \frac{d^2 z}{dx^2} \cdot \left(1 + \left(\frac{dz}{dx}\right)^2\right)^{-1.5}$$

Backup slides (8/8): parameters

Table 2 – Details of DCGAN architecture.

Layers	*G* model (deconvolutional)		*D* model (convolutional)	
	Number of kernels	Kernel size	Number of kernels	Kernel size
First layer	80	10	10	10
Second layer	40	10	20	10
Third layer	20	10	40	10
Fourth layer	10	10	80	10
Fifth layer	1	11	1	10

Table 3 – Parameters for the NSGA-II.

Population size	100
Max number of generations	250
Crossover	η_c = 15, rate = 0.9
Mutation	η_m = 15, rate = 0.01