CPSC 452: Project Proposal

Tristan Brigham, Eugene Han, Elder Veliz

March 29, 2024

1. **Problem Statement:**

In the domain of mechanical design, particularly in the aerospace industry, engineers aim to devise innovative system designs that not only meet but exceed existing aerodynamic performance benchmarks. However, these advancements often encounter substantial obstacles when transitioning from conceptual designs to tangible, real-world applications. Challenges such as material limitations in strength, flexibility, and environmental resistance, among others, obstruct the seamless realization of these sophisticated designs. As a result, engineers find themselves dedicating significant effort to modify these designs, ensuring they are viable for real-world application while often compromising on the originally targeted performance enhancements.

Efforts to refine this optimization process with Topological Optimization (TO) algorithms, including Solid Isotropic Material with Penalization (SIMP), level set, and density approaches, have faced limitations. These methods, while innovative, exhibit limitations in their ability to generalize beyond the initial designs provided by human engineers. They often result in oversimplified versions of the initial concept, lacking the nuanced features necessary to meet complex aerodynamic requirements. Consequently, the traditional design workflow, which combines human creativity with algorithmic precision to meet physical constraints, leads to extended design cycles that can span from several months to years.

Previous research leveraging Conditional Variational Autoencoders (cVAEs) (Yonekura and Sozuki 2021) has made strides in bridging the gap between aerodynamic performance objectives and the generation of corresponding airfoil shapes. While these methods mark significant progress, allowing for the exploration of new designs based on specified performance criteria, they are not without their limitations. Notably, the challenge lies in achieving an optimal balance between design fidelity and performance accuracy, with cVAEs sometimes struggling to produce designs that are both diverse and closely aligned with the stringent requirements of aerospace applications.

This backdrop sets the stage for our project, which proposes to pivot towards using Conditional Generative Adversarial Networks (cGANs) in the design of airfoil shapes. This approach is motivated by cGANs' superior capability for generating high-quality, diverse designs that more accurately adhere to specified aerodynamic constraints. Unlike cVAEs, which primarily focus on minimizing reconstruction and latent losses, cGANs introduce a robust adversarial training mechanism that directly aligns generated designs with physical and performance constraints through a discriminative feedback loop.

The core **objective** of this research is to use cGANs to create a more efficient design framework for airfoil shapes. By doing so, we aim to address design generalization challenges, multi-objective constraint integration, and designs optimized for advanced manufacturing techniques such as additive manufacturing. This research will focus on optimizing key aerodynamic performance metrics including lift, drag, and weight, under real-world constraints, thereby contributing to the development of aerospace vehicles with enhanced efficiency, performance, and environmental compatibility. With this preliminary study, we aim to establish a foundation for a new paradigm in swift, physics-driven design methodologies.

As an additional note, if time permits we will attempt to generalize the approach above into 3 dimensions, using physics engines to simulate the abilities of 3D designs. This will involve creating a cGAN similar to the airfoil cGAN which includes more parameters such as the target internal volume, expected mass, total lift, and drag.

We rely heavily on the following pieces of work for inspiration and guidance for how to complete the project:

(a) Yonekura & Suzuki 2021

(b) Almasri et. al. 2022

(c) Shu et. al. 2019

2. **Datasets:**

(a) *Custom Generated Datasets:* First, we will generate a dataset of simple airfoil designs using the NACA four-digit airfoil framework which can be trivially computed using online websites such as this one. We will convert the resulting designs from modulating the max camber, max camber position, and thickness while keeping the number of points constant into point clouds for further processing.

We are able to feed the XFOIL program point clouds that describe 2D the air foils generated above and the program is able to feed us metrics about how they perform in computational fluid dynamics (CFD) simulations such as the lift and drag of the design.

(b) *Department of Energy Airfoil Data:* Contains roughly 1800 airfoils that we can train on. It provides the shapes of different airfoils paired with labels specifying values such as the flow field values (momentum, energy, and vorticity) and summary values (coefficients of lift, drag, and momentum) for the designs.

(c) *UIUC Airfoil Coordinates Database:* Has 1600 airfoils parameterized by their coordinates in a format that can be easily passed to XFOIL to compute lift and drag amongst other values. It will provide a useful resource for initial airfoil designs that the model can learn.

(d) *ShapeNET:* This database provides us with high quality scans over 52,000 different items including passenger jets, cars, and boats which can be transformed into voxels or point fields to be used as training data for the GAN. We will transform the data that are relevant to aerodynamics optimization into a 2D representation if we perform experiments beyond simple airfoil optimization.

Additionally, if we have time to complete a 3D version of the model, this dataset will serve as the basis for said models.

3. **Model Architecture:**

We will use a **Conditional Generative Adversarial Network** (cGAN) architecture to generate airfoils, aiming to adhere to specific aerodynamic performance parameters.

The **generator** will take as input a combination of noise and specific design parameters (e.g. lift coefficient, drag reduction, etc,). It will output a point cloud representing the airfoil shape that theoretically satisfies the design parameter set. One can see an example of such a representation of an airfoil below where each point is concatenated into a larger vector that will serve as input for the model. It is crucial to note that we will attempt to be as consistent as possible with respect to keeping the orientation and relative position of points constant across the dataset.

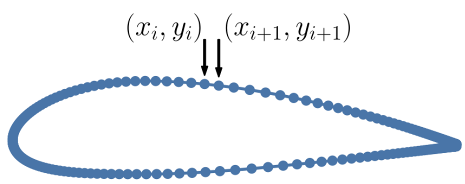

Figure 1: Example airfoil from Yonekura & Suzuki 2021 which we rely on for the initial idea

The **discriminator** will comprise of multiple networks, each dedicated to evaluating compliance with a specific aerodynamic parameter. We will integrate attention mechanisms within these networks to focus on relevant features of the generated designs, thereby enhancing the evaluation of parameters. In addition to these parameter-specific networks, a traditional discriminator, also enhanced with attention mechanisms, will focus on the broader task of distinguishing between real and generated designs. This network ensures that, beyond meeting specific aerodynamic parameters, the designs also possess the qualities and characteristics of feasible airfoil designs.

The adversarial loss function $L(G, D)$, which applies to the generator and the discriminator, is:

$$L(G, D) = \min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x|c)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|c)))] \tag{1}$$

where $G((z|c), \theta_g)$ is the conditional generator with θ_g representing its parameters, $D((x|c), \theta_d)$ is the conditional generator with θ_d representing its parameters, z is the latent vector, x is the design variables vector, and c is the vector of the m specific input conditions (aerodynamic metrics).

The objective functions for the generator $\mathcal{L}_{\text{G}}$ and traditional discriminator $\mathcal{L}_{\text{D}}$ are represented as:

$$\mathcal{L}_{\text{G}} = \mathcal{L}_{\text{reconstruction}} + \lambda_{\text{adversarial}}(\mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|c)))]) + \sum_{j=1}^{m} \mathcal{L}_j \tag{2}$$

$$\mathcal{L}_{\text{D}} = -\mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x|c)] - \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|c)))] \tag{3}$$

where $\mathcal{L}_{\text{reconstruction}} = \frac{1}{N}||x_i - \hat{x}_i||_2^2$ guides the generated designs towards an output that would resemble a real-life airfoil (where N is the batch size, x_i is a design from the training data set, and $\hat{x}_i$ is a design generated by the generator), the adversarial loss measures the ability of the generator to produce designs that are realistic enough to be classified as real by the discriminator (and it is weighted by $\lambda_{\text{adversarial}}$ to fine-tune the impact of the adversarial dynamics on the total loss, balancing the drive for realistic design generation against the precision of parameter-specific optimizations.), and each $\mathcal{L}_j = \|c_j - \hat{c}_j\|_1$ corresponds to losses associated with each of the m specific aerodynamic performance metrics — drawing inspiration from Almasri et. al. 2022.

During the **pre-processing** stage, data preparation will involve cleaning and standardizing existing design data from the Department of Energy and UIUC. Each airfoil design will be discretized into a set of coordinate points $\{(x_i, y_i) \mid i = 1, 2, \ldots, N\}$, and transformed into a vector form $s = \{x_1, y_1, x_2, y_2, \ldots, x_N, y_N\}$, as done in Yonekura & Suzuki 2021. This vector representation will then be associated with a vector of aerodynamic performance indices, which can encompass discrete or continuous variables such as lift coefficient, drag coefficient, and moment coefficient.

To acquire these aerodynamic metrics, each airfoil's vector representation will be subjected to a series of XFOIL simulations. These simulations will be standardized to maintain uniformity across the dataset and will capture the aforementioned parameters at various angles of attack to thoroughly characterize the airfoil's performance.

The vectorized airfoil shapes along with their aerodynamic performance metrics will be extracted and structured into a format conducive to our model architecture. If necessary, we will augment the dataset to enhance its robustness; Almasri et. al. 2022, for example, augmented the training dataset using rotations of 90°, 180°, and 270°. The augmented data, together with the original dataset, will serve both as the inputs to the model and the labels for the data, with the aerodynamic parameters (e.g. lift coefficient, drag coefficient, etc.) serving as our ground truth.

We will partition this final dataset into training, validation, and testing data.

During the **training** process, the generator will receive an input composed of noise and aerodynamic parameters derived from XFOIL simulations. This information will guide the generator in generating airfoil designs that aim to meet specified performance benchmarks. (Obviously, initially, these designs may not meet the specified performance criteria accurately.)

The discriminator ensemble, which includes both the traditional discriminator for assessing the overall authenticity and parameter-specific discriminators, will then evaluate the generated designs. The discriminator will be presented with actual airfoil designs from our dataset (i.e. "real" samples) as well as the designs generated by the generator ("fake" or "generated" samples). By evaluating both, the discriminator will learn to identify the characteristics that differentiate real designs from those that are generated. It will attend to critical aspects of the design that impact the specified aerodynamic parameters, providing feedback on each aspect of performance.

During the **feedback loop**, the combined set of discriminators — encompassing the traditional discriminator alongside specific parameter discriminators — will provide gradient-based feedback to the generator. This feedback consists of gradients that reflect discrepancies between the generated designs and the set of specific aerodynamic parameters. We will implement this critique by back-propagating the errors from all discriminators to the generator. The gradients specifically highlight which parameters (e.g., lift, drag) are not within the desired range and suggest the direction of adjustment.

The generator will adjust its parameters based on the discrinator's critique, aiming to produce airfoil designs that better align with the specified aerodynamic performance criteria. Through this cycle of generation, feedback, and adjustment that repeats over multiple epochs, the generator will (ideally) progressively improve its ability to produce designs satisfying the discriminator's evaluations.

We will continue training until the model converges, indicated by a stabilization in the loss functions and the generator's increasing ability to produce airfoil designs that meet the specified aerodynamic criteria. We will likely have to adjustment the learning rate and the $\lambda_{adversarial}$ parameter fine-tune the model's performance using a validation set.

Almasri et. al. 2022, which conducted a similar experiment, has the following structure:

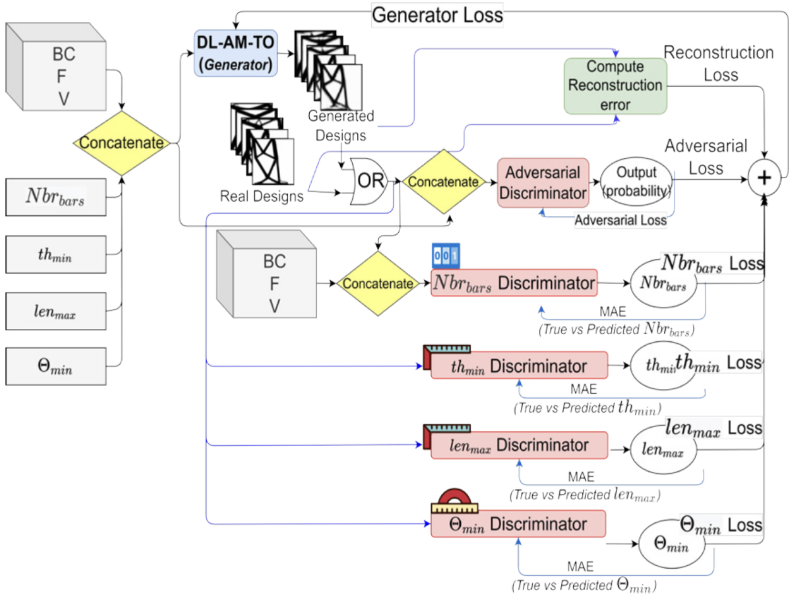

Figure 2: Example network from Almasri et. al. 2022 that performs a similar experiment

4. **Problems We Will Likely Face:**

One of the largest problems with any GAN is always going to be the training time that it takes for the model to converge. We expect that we will face similar problems in this situation because we are dealing with highly complex data which will only compound the amount of time that it takes to train the model on top of what is already a computationally intensive model. We will mitigate this by ensuring that every run we complete of the model gleans new insights for us and will leverage accelerators such as GPU clusters at Yale.

Additionally, neural networks can be rather sensitive to the orientation of points in some relatively constant surface with respect to the inputs. We need to make sure that the points forming the point cloud are generated by the neural network in a somewhat consistent manner such that we can easily construct a surface specifying the airfoil from the output points.

Converting the output of our model to something that can be evaluated by XFOIL could also pose challenges given the structure of XFOIL data being different and somewhat more specific than the output of our model.

5. **Evaluation of Results:**

   (a) *Qualitative:* Qualitatively, we can inspect the generated designs to manually check whether the model is converging to designs that are similar to real modern-day airfoil designs. For instance, if we are trying to design airfoils with some characteristics and the model is generating squares, we know that we are having poor convergence in the model. On the other hand, if the designs roughly mimic modern airfoil designs, we know that the model is converging.

   (b) *Quantitative:* Quantitatively, we can measure several factors to evaluate how well the model generates airfoil designs. First, we can run XFOIL to check that the metrics we are discriminating on (e.g. drag, lift, etc.) of the generated airfoils are close to the values we want. Then, we compute the mean squared error (MSE) between the coefficients of drag, lift, etc. of the target and generated designs. Moreover, we will also analyze the raw differences and, similarly, the target vs. actual design metrics to determine if consistent under/overshooting exists.

       We can also examine the variation in the generated designs to ensure the model is not producing identical or trivial solutions. We can measure diversity by computing the variance in the generated designs' shape features.