Tailoring Generative Adversarial Networks for Smooth Airfoil Design

Joyjit Chattoraj
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
joyjitc@ihpc.a-star.edu.sg

Jian Cheng Wong
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
wongj@ihpc.a-star.edu.sg

Zhang Zexuan
School of Continuing and Lifelong Education
National University of Singapore
Singapore 119260, Republic of Singapore
E0954403@u.nus.edu

Manna Dai
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
manna_dai@ihpc.a-star.edu.sg

Xia Yingzhi
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
xia_yingzhi@ihpc.a-star.edu.sg

Li Jichao
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
li_jichao@ihpc.a-star.edu.sg

Xu Xinxing
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
xuxinx@ihpc.a-star.edu.sg

Ooi Chin Chun
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
ooicc@cfar.a-star.edu.sg

Yang Feng
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
yangf@ihpc.a-star.edu.sg

Dao My Ha
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
daomh@ihpc.a-star.edu.sg

Liu Yong
Institute of High Performance Computing
Agency for Science, Technology and Research
Singapore 138632, Republic of Singapore
liuyong@ihpc.a-star.edu.sg

***Abstract*—In the realm of aerospace design, achieving smooth curves is paramount, particularly when crafting objects such as airfoils. Generative Adversarial Network (GAN), a widely employed generative AI technique, has proven instrumental in synthesizing airfoil designs. However, a common limitation of GAN is the inherent lack of smoothness in the generated airfoil surfaces. To address this issue, we present a GAN model featuring a customized loss function built to produce seamlessly contoured airfoil designs. Additionally, our model demonstrates a substantial increase in design diversity compared to a conventional GAN augmented with a post-processing smoothing filter.**

***Index Terms*—GAN, generative AI, custom loss, airfoil design**

I. Introduction

Generative AI models, such as Generative Adversarial Network (GAN) and Variational Autoencoder (VAE), have demonstrated promising outcomes in engineering design. Applications span a range of fields, encompassing structural optimization, materials design, and shape synthesis [1]. In recent years, particularly in the domain of airfoil design [2], numerous variations of GAN models have been investigated to discover novel shapes capable of achieving optimal aerodynamic performance across a diverse set of operating conditions [3]–[7].

However, it is frequently reported in the literature that GAN has limitations generating smooth airfoil curves [8]–[11]. A commonly employed strategy to address this issue involves integrating a smoothing filter as a post-processing technique [8].

Chen et al [12]–[14] proposed a novel model, BézierGAN, to avoid post-processing of airfoil curves, instead, they employed a smoothing method, in this case, Bézier curve, as an integral part of the GAN generator. During GAN training, the model learns the optimized values for Bézier parameters. In addition, a set of regularization terms has been imposed that enhances the complexity of the model, which could potentially limit the model's generalizability across various applications.

Yonekura et al [15] used Conditional Wasserstein GAN (CWGAN-GP) and found that the model can generate smooth airfoil curves. Notably, WGAN was originally proposed to tackle the vanishing gradient problem in GAN models [16]. Hence, whether the primary reason for non-smoothness lies in a vanishing gradient remains an open question. Tan et al [9] with a similar CWGAN-GP model could not generate smooth airfoil curves consistently. One noticeable difference between these two models is that Tan et al [9] considered more than one parameter for airfoil design including a shape-determined parameter (area).

Wang et al [17] achieved smooth airfoil shapes by integrating a VAE and a GAN, referred to as VAEGAN, originally proposed in [18]. Here, VAE acts as a GAN generator, and it can generate smooth curves without imposing any explicit methodology for curve smoothness.

In our proposed methodology, we attempt to tackle the

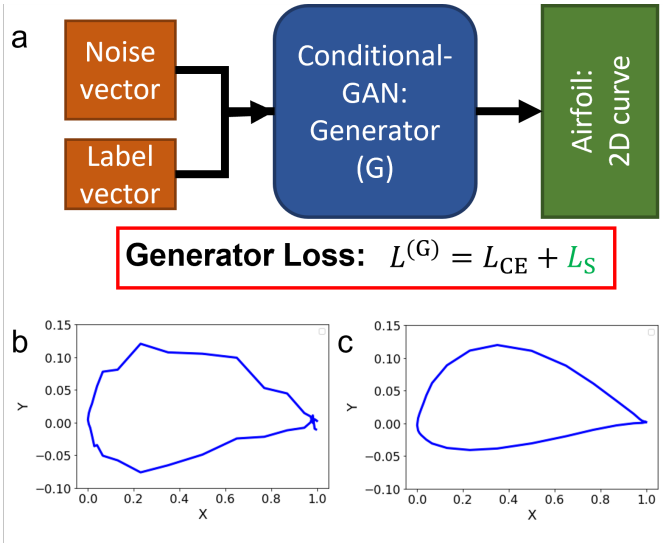

Fig. 1. (a) The generator (G) of a trained conditional-GAN takes a noise vector and a label vector as inputs and the generator outputs a 2D airfoil curve. The generator is trained with a loss function $L^{(G)}$ which includes a binary cross entropy loss (L_{CE}) and a mean square error loss L_S to impose curve smoothing. (b) The generator outputs non-smooth curves if $L^{(G)}$ only includes L_{CE}, i.e., without L_S. (c) The generator outputs desired smooth curves when L_S is included in the generator loss in addition to L_{CE}.

non-smoothness problem of GAN explicitly. We propose a custom loss function for the generator model of conditional GAN (Fig. 1a). This loss function calculates a mean square error if a generated curve deviates from its moving average position. The custom loss function improves the ability of the GAN model to transition from generating non-smooth curves to creating smooth airfoil curves, as illustrated in Fig. 1b,c. Moreover, we find that the custom loss facilitates the GAN model to generate more diverse airfoil shapes than the original GAN without the custom loss function and incorporated with a smoothing filter as a post-processing technique. As the custom loss requires computing only the moving average, this approach is generalizable and can be easily adopted in other engineering designs to impose smoothness.

The remaining sections of this article are organized as follows: (i) Dataset: Here, we provide an overview of the airfoil data utilized for training the GAN models. (ii) Methodology: This section delves into the specifics of the GAN architecture, custom loss, and the training procedure. (iii) Results: We analyze the effectiveness of the custom loss function in this segment. (iv) Conclusions: Finally, we summarize the main findings and discuss potential avenues for future research.

II. Dataset

Our dataset consists of 1399 airfoils which are obtained from the UIUC Airfoil Data Site [19] using an open-source pre-processing code [20], [21]. These airfoils are essentially 2D curves with a set of X and Y coordinates, where X coordinates strictly range from 0 to 1. The Y coordinates are flexible and typically vary between -0.3 to 0.4 (Fig. 2). The maximum thickness (τ), i.e., measured perpendicular to the

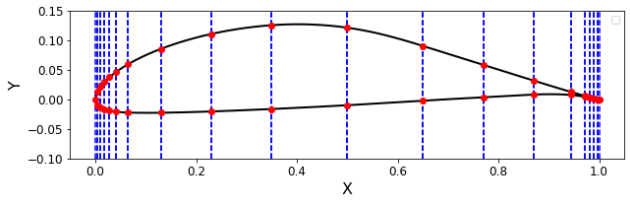

Fig. 2. An original airfoil (solid line) after reducing the number of points to 38 (symbols).

camber line of an airfoil, varies between 0.04 and 0.38 with a median value of 0.12.

Subsequently, these airfoils underwent assessment under three distinct operating conditions: Reynolds number (Re), Mach number (M), and angle of attack (α). The airfoil's aerodynamic performance was evaluated by calculating the lift coefficient (c_l) and drag coefficient (c_d) utilizing XFOIL [22]. The numerical simulation generated a comprehensive dataset comprising 654,963 samples, averaging 468 samples per airfoil, encompassing different operating conditions, i.e., Re $(2000000 - 12000000)$, M $(0 - 0.3)$, and α $(0 - 33)$, with their calculated c_l and c_d.

TABLE I
8 Classes and Respective Number of Airfoils for Training

Class	c_l/c_d	α	τ	Airfoils
000	low	low	low	740
001	low	low	high	627
010	low	high	low	742
011	low	high	high	649
100	high	low	low	739
101	high	low	high	633
110	high	high	low	717
111	high	high	high	566

In essence, our GAN design challenge is framed as a classification problem, where the objective is to create airfoil shapes that align with a particular class. We define 8 classes to create airfoil shapes from 3 parameters: τ representative of airfoil geometry, α representative of operating conditions, and the ratio c_l/c_d representative of aerodynamic performance (Table I). A class is characterized by combinations of low and high values of τ, α, and c_l/c_d. The thresholds for distinguishing between low and high values are established based on the median values of these parameters, which are 0.12, 10, and 100, respectively. We proceed to create a training dataset by determining the class to which each airfoil belongs. Note that while an airfoil may be associated with multiple classes, its presence in a class is considered only once. We find that the number of airfoils varies between 566 and 742 in 8 classes, a total of 5413 samples, implying a reasonably balanced dataset for GAN training.

III. Methodology

A. Pre-processing

We employ a pre-processing methodology to standardize airfoil coordinates for GAN training. This process ensures uni-

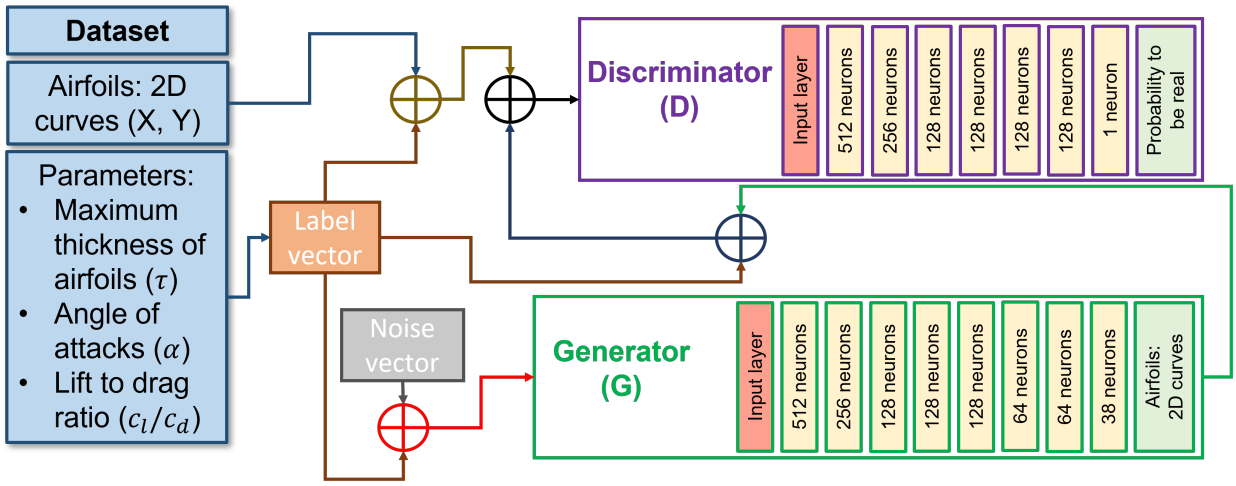

Fig. 3. The architecture of conditional-GAN consists of a generator (G) and a discriminator (D). Both G and D are fully connected deep neural networks.

form X coordinates across all airfoils, maintaining consistency in their sequence, while strategically reducing the number of points. Emphasis is placed on critical airfoil segments, such as the leading and trailing edges. Employing a cosine distribution, we set 20 X coordinates; excluding the two endpoints (X=0 and X=1), the remaining 18 X coordinates each have 2 corresponding Y coordinates for the upper and lower surfaces (Fig. 2). Consequently, a total of 38 Y coordinates distinguish one airfoil from another, and these 38 Y coordinates serve as inputs for GAN training.

B. GAN architecture

We adopt the conditional GAN technique [23] which comprises a generator (G) and a discriminator (D), depicted in Fig. 3. G is a fully connected deep neural network with the rectified linear unit (ReLU) as an activated function. G receives an input vector formed by concatenating a random noise vector (128 dimensions, Gaussian noise with mean 0 and standard deviation 1) and a label vector (6 dimensions). Label vector represents the 8 classes (Table I), with an example being 100101 label vector for class-011. G creates 38 Y coordinates belonging to the 2D curve of an airfoil.

D receives an input vector formed by concatenating the real coordinates (38 dimensions) and generated coordinates (38 dimensions) from G associated with the label vector (6 dimensions). D determines the probability of the generated coordinates to be real. D is also a fully connected deep neural network; the last layer of D uses the sigmoid activation function, and all the other neurons use ReLU.

C. Custom loss function

To enforce smoothness on the 2D curve of an airfoil, we tailor the generator training by customizing the loss function $L^{(\mathrm{G})}$ as

$$L^{(\mathrm{G})} = L_{\mathrm{CE}} + L_{\mathrm{S}}, \tag{1}$$

where L_{CE} is the standard binary cross-entropy loss, whereas L_{S} introduces a penalty for any deviation of the generated Y coordinates (with dimensions $n_c = 38$) from their respective moving average points. We compute the moving average $\bar{Y}_i$ for all Y coordinates with a window of 3 points as

$$\bar{Y}_i = \frac{1}{3}\sum_{j=-1}^{1} Y_{i+j}, \tag{2}$$

where $i = 1, 2, \ldots, n_c$, and the index $i + j$ iterates through 1 and n_c in a cyclic manner. The coordinates are arranged in a loop where $i = 1$ corresponds to the endpoint at $X_i = 1$, $i = 2$ corresponds to the upper surface point at $X_i = 0.996$, and $i = n_c$ corresponds to the lower surface point at $X_i = 0.996$. The deviation Δ_i between a Y coordinate Y_i and its moving average point is then

$$\Delta_i = Y_i - \bar{Y}_i. \tag{3}$$

We define the smoothing loss L_{S} as the mean square deviation of Δ_i as shown below

$$L_{\mathrm{S}} = \frac{\omega}{n_c}\sum_{i=1}^{n_c} \Delta_i^2, \tag{4}$$

where the coefficient ω determines the strength of L_{S} with respect to L_{CE}. For the discriminator, we use the binary cross-entropy loss function only.

D. GAN training

The generator (G) and discriminator (D) are trained simultaneously in a competitive manner incorporating the custom loss function (1). In a training loop, G creates 2D airfoil curves, and D evaluates both real and generated airfoils without knowing the source. For both G and D, we use ADAM optimizer and the learning rate 10^{-4}. We set epochs at $30,000$. During training we observed that beyond the first several thousand epochs, both the G-loss and D-loss converged to approximately the same value, maintaining a steady-state until the end of the training process. We experimented with three distinct values for ω in (4): 1, 10, and 100. Our findings indicated that setting ω to 10 resulted in optimal smoothing for the airfoils.

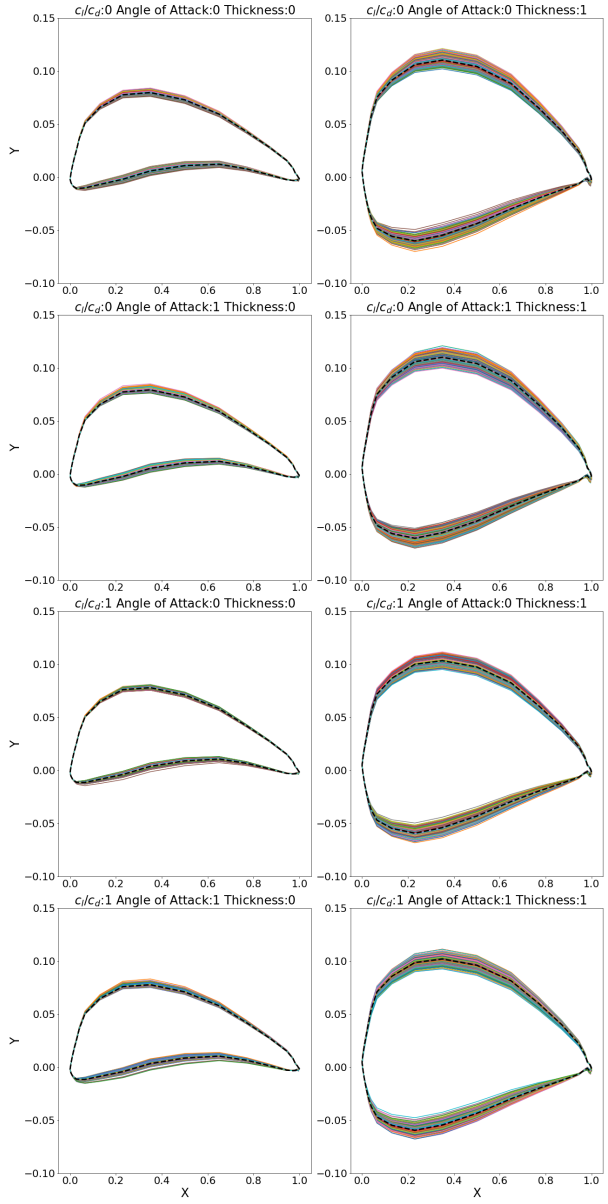

Fig. 4. Generated airfoil samples from the GAN trained without the smoothing loss and augmented with a post-processing smoothing filter. Each panel represents a class defined by low/high (0/1) c_l/c_d, α, and τ. It displays 600 airfoil samples and their mean shape (dashed curve).

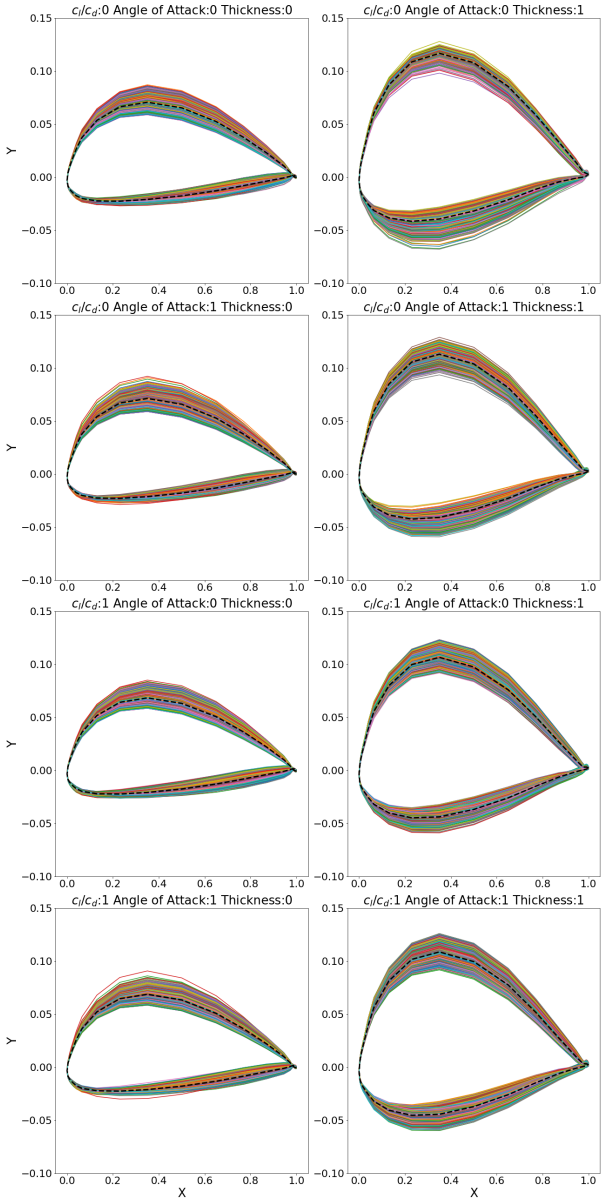

Fig. 5. Generated airfoil samples from the GAN model trained with the smoothing loss. Each panel represents a class defined by low/high (0/1) c_l/c_d, α, and τ. It displays 600 airfoil samples and their mean shape (dashed curve).

IV. Results

To evaluate the efficacy of our proposed custom loss function (1), we conducted training experiments with two GAN models: one without a smoothing loss and another incorporating the smoothing loss function, to be referred to as smoothGAN, henceforth. We then compared their performances in generating airfoils. A total of 600 airfoil samples were generated for each of the 8 classes, and we assessed these samples using three statistical metrics:

1) $\text{ACC}^{(\tau)}$: Accuracy that measures the percentage of the total airfoils possessing the correct (low/high) thickness at a given class.
2) $\sigma^{(\tau)}$: Diversity in thickness, calculated as the standard deviation of thickness after being rescaled by their maximum observed value.
3) S: Diversity in shape, following [24], it is calculated as the average distance of the mean shape $Y^{(\text{m})}$ from all the generated samples $\left\langle \sqrt{\sum_{i=1}^{n_c}(Y_i - Y_i^{(\text{m})})^2} \right\rangle_{\text{samples}}$.

A. GAN without smoothing loss function

We found that GAN which is trained without the smoothing loss function always generated non-smooth 2D curves, an example can be seen in Fig. 1b. A Savitzky-Golay filter [25] is further augmented with the GAN model as a post-processing

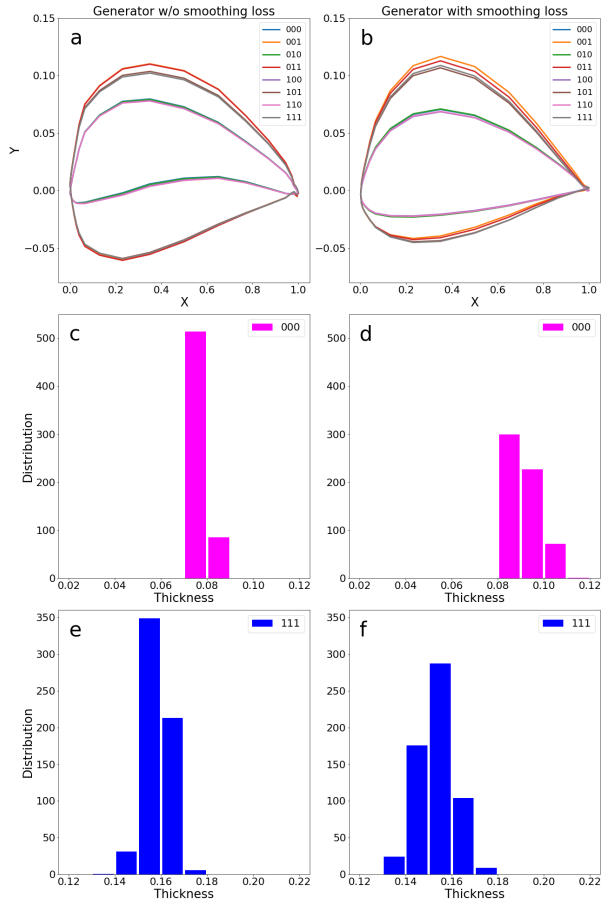

Fig. 6. Comparison of airfoils which are generated from GAN trained without smoothing loss and augmented with a post-processing smoothing filter (3 panels in left column), and GAN trained with smoothing loss (3 panels in right column). (a) and (b) show the mean shape of 600 airfoils, a total of 8 shapes corresponding to 8 classes. Distribution of τ of 600 airfoils for a fixed class: (c) and (d) for class-000, and (e) and (f) for class-111.

technique that smooths out the GAN-generated samples, which are displayed in Fig. 4. We illustrate the mean shape of airfoils for each class in Fig. 6a to showcase the shape diversity among classes. Additionally, the distribution of τ is depicted in Fig. 6c,e as an alternative measure of geometric diversity.

B. smoothGAN: GAN with smoothing loss function

The smoothGAN model trained with the custom loss function generates smooth 2D curves, which do not require any post-processing. 600 samples for each of 8 classes, a total of 4800, plotted in Fig. 5, all exhibits smooth shapes. It is evident in Fig. 5 and also in Fig. 6b,d,f that in comparison to GAN, smoothGAN achieves more diversity in generated shapes. We further quantitatively demonstrate the diversity through two metrics: $\sigma^{(\tau)}$ and S.

The comparative performance of these two GAN models is summarized in Table II. Both models achieve a 100% accuracy ($\text{ACC}^{(\tau)}$), indicating that the generated airfoils consistently exhibit the correct (high/low) thickness corresponding to their respective classes. However, smoothGAN exhibits notably

TABLE II
PERFORMANCE OF GAN

Class	w/o Smoothing Loss			with Smoothing Loss		
	$\text{ACC}^{(\tau)}(\%)$	$\sigma^{(\tau)}$	S	$\text{ACC}^{(\tau)}(\%)$	$\sigma^{(\tau)}$	S
000	100	0.007	0.003	100	0.057	0.017
001	100	0.024	0.009	100	0.044	0.017
010	100	0.007	0.003	100	0.061	0.018
011	100	0.028	0.01	100	0.036	0.02
100	100	0.005	0.003	100	0.05	0.017
101	100	0.027	0.009	100	0.044	0.02
110	100	0.008	0.003	100	0.057	0.018
111	100	0.03	0.009	100	0.043	0.02

elevated values for both $\sigma^{(\tau)}$ and S. As seen in Fig. 6, distribution of τ is broader for smoothGAN than the former GAN leading to approximately 1.5 to 10 times higher $\sigma^{(\tau)}$ values. Similarly, smoothGAN exhibits approximately 2 to 6 times higher S values than the former GAN. It is noteworthy that the minimum value of S is zero implying that the model generates only one airfoil shape. This situation may occur in the presence of mode collapse, a common drawback associated with GANs [26].

V. CONCLUSIONS

In this study, we developed a methodology to generate smooth airfoils using GAN. We showed that conditional GAN has limitations in generating smooth curves, supporting the previous findings reported in the literature. Typically, this shortcoming has been addressed by augmenting a smoothing filter with GAN as a post-processing technique. We demonstrated explicitly that employing such a post-processing technique can attain high accuracy. Nevertheless, this approach may generate a restricted range of airfoil shapes, and the absence of diversity in shape can constrain the GAN's ability to explore novel designs.

Our proposed methodology integrates a custom loss function with GAN, comprising a smoothing loss that penalizes deviations of the generated shape from the reference shape assessed through a moving average. This approach enhances the GAN's capability to consistently generate smooth airfoils, with 2 to 10 times greater diversity compared to the original GAN.

In future applications, we plan to implement our proposed GAN methodology in various scenarios to generate smooth curves and surfaces. One potential application could be in the realm of 3D airfoil design.

ACKNOWLEDGMENT

This research is supported by Agency for Science Technology and Research (A*STAR) under the AME Programmatic project: Explainable Physics-based AI for Engineering Modelling & Design (ePAI) [Award No. A20H5b0142].

REFERENCES

[1] L. Regenwetter, A. H. Nobari, and F. Ahmed, "Deep generative models in engineering design: A review," *Journal of Mechanical Design*, vol. 144, no. 7, p. 071704, 2022.

[2] J. Li, X. Du, and J. R. Martins, "Machine learning in aerodynamic shape optimization," *Progress in Aerospace Sciences*, vol. 134, p. 100849, Oct. 2022.

[3] J. Li, M. Zhang, J. R. Martins, and C. Shu, "Efficient aerodynamic shape optimization with deep-learning-based geometric filtering," *AIAA journal*, vol. 58, no. 10, pp. 4243–4259, 2020.

[4] E. Yilmaz and B. German, "Conditional generative adversarial network framework for airfoil inverse design," in *AIAA aviation 2020 forum*, 2020, p. 3185.

[5] A. Heyrani Nobari, W. Chen, and F. Ahmed, "Pcdgan: A continuous conditional diverse generative adversarial network for inverse design," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 606–616.

[6] J. Li and M. Zhang, "On deep-learning-based geometric filtering in aerodynamic shape optimization," *Aerospace Science and Technology*, vol. 112, p. 106603, May 2021.

[7] Q. Du, T. Liu, L. Yang, L. Li, D. Zhang, and Y. Xie, "Airfoil design and surrogate modeling for performance prediction based on deep learning method," *Physics of Fluids*, vol. 34, no. 1, 2022.

[8] G. Achour, W. J. Sung, O. J. Pinon-Fischer, and D. N. Mavris, "Development of a conditional generative adversarial network for airfoil shape optimization," in *AIAA Scitech 2020 Forum*, 2020, p. 2261.

[9] X. Tan, D. Manna, J. Chattoraj, L. Yong, X. Xinxing, D. M. Ha, and Y. Feng, "Airfoil inverse design using conditional generative adversarial networks," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2022, pp. 143–148.

[10] G. B. Santos, A. V. Pantaleão, and L. O. Salviano, "Using deep generative adversarial network to explore novel airfoil designs for vertical-axis wind turbines," *Energy Conversion and Management*, vol. 282, p. 116849, 2023.

[11] K. Wada, K. Suzuki, and K. Yonekura, "Physics-guided training of gan to improve accuracy in airfoil design synthesis," *arXiv preprint arXiv:2308.10038*, 2023.

[12] W. Chen and M. Fuge, "B\'eziergan: Automatic generation of smooth curves from interpretable low-dimensional parameters," *arXiv preprint arXiv:1808.08871*, 2018.

[13] W. Chen, K. Chiu, and M. Fuge, "Aerodynamic design optimization and shape exploration using generative adversarial networks," in *AIAA Scitech 2019 Forum*, 2019, p. 2351.

[14] W. Chen, K. Chiu, and M. D. Fuge, "Airfoil design parameterization and optimization using bézier generative adversarial networks," *AIAA journal*, vol. 58, no. 11, pp. 4723–4735, 2020.

[15] K. Yonekura, N. Miyamoto, and K. Suzuki, "Inverse airfoil design method for generating varieties of smooth airfoils using conditional wgan-gp," *Structural and Multidisciplinary Optimization*, vol. 65, no. 6, p. 173, 2022.

[16] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[17] Y. Wang, K. Shimada, and A. B. Farimani, "Airfoil gan: Encoding and synthesizing airfoils for aerodynamic shape optimization," *Journal of Computational Design and Engineering*, p. qwad046, 2023.

[18] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.

[19] M. S. Selig. (1996) Uiuc airfoil data site. [Online]. Available: https://m-selig.ae.illinois.edu/ads/coord_database.html

[20] J. Li, S. He, and J. R. Martins, "Data-driven constraint approach to ensure low-speed performance in transonic aerodynamic shape optimization," *Aerospace Science and Technology*, vol. 92, p. 536–550, Sep. 2019.

[21] J. Li, M. A. Bouhlel, and J. R. Martins, "Data-based approach for fast airfoil analysis and optimization," *AIAA Journal*, vol. 57, no. 2, pp. 581–596, 2019.

[22] M. Drela, "Xfoil: An analysis and design system for low reynolds number airfoils," in *Low Reynolds Number Aerodynamics: Proceedings of the Conference Notre Dame, Indiana, USA, 5–7 June 1989*. Springer, 1989, pp. 1–12.

[23] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[24] N. C. Brown and C. T. Mueller, "Quantifying diversity in parametric design: a comparison of possible metrics," *AI EDAM*, vol. 33, no. 1, pp. 40–53, 2019.

[25] W. H. Press and S. A. Teukolsky, "Savitzky-golay smoothing filters," *Computers in Physics*, vol. 4, no. 6, pp. 669–672, 1990.

[26] D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee, "Diversity-sensitive conditional generative adversarial networks," *arXiv preprint arXiv:1901.09024*, 2019.