DiffAirfoil: An Efficient Novel Airfoil Sampler Based on Latent Space Diffusion Model for Aerodynamic Shape Optimization

Zhen Wei [*], Edouard Dufour [†], Colin Pelletier [‡], and Pascal Fua [§]
EPFL, Lausanne, 1015, Switzerland

Michaël Bauerheim [¶]
ISAE-SUPAERO, Toulouse, 31055, France

Surrogate-based optimization is widely used for aerodynamic shape optimization, and its effectiveness depends on representative sampling of the design space. However, traditional sampling methods are hard-pressed to effectively sample high-dimensional design spaces. This paper introduces *DiffAirfoil*, a new airfoil sampling method based on diffusion in an automatically learned latent space. *DiffAirfoil* is highly data-efficient and requires significantly fewer training geometries than Generative Adversarial Networks. It ensures the validity of sampled airfoils through an automatic parameterization. We demonstrate *DiffAirfoil*'s capability to generate diverse and valid 2D airfoil shapes, while also facilitating conditional generation without the need for adaptation or retraining. Comprehensive benchmarks show that our method offers significant advantages in data efficiency, ease of implementation, and adherence to conditions. Therefore, *DiffAirfoil* presents a promising approach to enhancing sampling efficiency for surrogate models in aerodynamic shape optimization tasks.

Nomenclature

$C, \mathbf{c}$	=	the geometric condition and the geometric pattern measurement
$\mathbf{c}_A, \mathbf{c}_{MT}$	=	the measurements for airfoil's area and maximum thickness
C_L, C_D	=	the lift and drag coefficients
D^k_{intra}, D^k_{inter}	=	the k-intra-sample distance and k-inter-sample distance
d	=	the dimension of the latent space
f_Θ, Θ	=	the Latent Space Model and its weights
$\mathcal{F}$	=	the Fréchet inception distance
g_Φ, Φ	=	the Latent Space Diffusion Model and its weights
h^G_Γ, Γ	=	the generator of the generative adversarial network and its weights
h^D_Π, Π	=	the discriminator of the generative adversarial network and its weights
L	=	the loss function
M, V, E	=	the airfoil contour and its vertices, edges
p, q_Φ	=	the forward and reverse pass of the Latent Space Diffusion Model
S	=	the realistic airfoil geometry from dataset
sm	=	the surrogate model
T	=	the time step of diffusion model
$\mathbf{Z}, \mathbf{z}$	=	the airfoil latent space and a latent code

I. Introduction

Surrogate-based optimization offers a simple and effective approach to aerodynamic shape optimization, eliminating the need to derive and implement an adjoint solver for each problem. It relies on a surrogate model to emulate the

[*]Doctoral Student, Computer Vision Lab, School of Computer and Communication Sciences, AIAA Student Member.
[†]Master Student in Microtechnics.
[‡]Master Student in Communication Systems.
[§]Professor, Computer Vision Lab, School of Computer and Communication Sciences
[¶]Associate Professor, Department of Aerodynamics, Energies and Propulsion

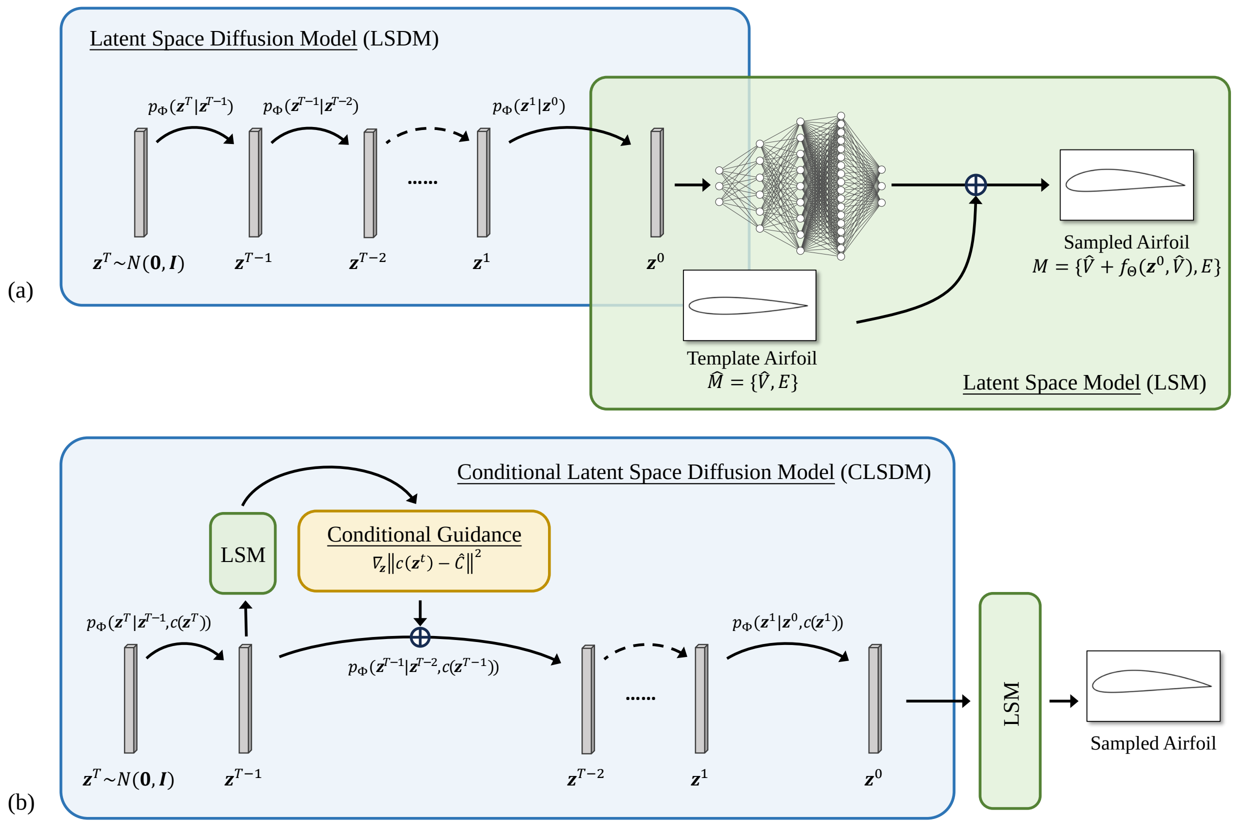

Latent Space Diffusion Model (LSDM)
$p_\Phi(\mathbf{z}^T|\mathbf{z}^{T-1})$
$p_\Phi(\mathbf{z}^{T-1}|\mathbf{z}^{T-2})$
$p_\Phi(\mathbf{z}^1|\mathbf{z}^0)$
$\mathbf{z}^T \sim N(\mathbf{0}, \mathbf{I})$
$\mathbf{z}^{T-1}$
$\mathbf{z}^{T-2}$
$\mathbf{z}^1$
$\mathbf{z}^0$
Sampled Airfoil
$M = \{\hat{V} + f_\Theta(\mathbf{z}^0, \hat{V}), E\}$
Template Airfoil
$\hat{M} = \{\hat{V}, E\}$
Latent Space Model (LSM)
(a)
Conditional Latent Space Diffusion Model (CLSDM)
LSM
Conditional Guidance
$\nabla_z \| c(\mathbf{z}^t) - \hat{c} \|^2$
$p_\Phi(\mathbf{z}^T|\mathbf{z}^{T-1}, c(\mathbf{z}^T))$
$p_\Phi(\mathbf{z}^{T-1}|\mathbf{z}^{T-2}, c(\mathbf{z}^{T-1}))$
$p_\Phi(\mathbf{z}^1|\mathbf{z}^0, c(\mathbf{z}^1))$
$\mathbf{z}^T \sim N(\mathbf{0}, \mathbf{I})$
$\mathbf{z}^{T-1}$
$\mathbf{z}^{T-2}$
$\mathbf{z}^1$
$\mathbf{z}^0$
LSM
Sampled Airfoil
(b)

Fig. 1 ***DiffAirfoil* framework. (a) Unconditional *DiffAirfoil* and (b) Conditional *DiffAirfoil*.**

behavior of a simulator, thereby accelerating the exploration of the shape space [1–5]. The key to their effectiveness lies in the ability to train the surrogate model using a representative set of shapes that cover potential optimization trajectories in the design space.

Traditional sampling methods, such as Latin hypercube sampling (LHS) [6], have limitations in sampling high-dimensional design spaces. They often depend on low-dimensional, handcrafted parameterizations, which restrict design space flexibility. Additionally, traditional methods struggle to guarantee the generation of realistic samples, resulting in inefficient sampling and the need for substantial human intervention during post-processing. Recently, there has been much interest in using the Generative Adversarial Networks (GANs) [7] to generate 2D airfoils in a data-driven manner [8–12]. Despite some successes, GANs are not ideal for rapid and cost-effective implementation in aerodynamic applications due to the challenges inherent to adversarial training: First, GAN's training requires a large dataset to ensure stability and prevent mode collapse, a catastrophic occurrence that stops GANs from producing diverse samples. Even in the comparatively simple case of 2D airfoils, the training sets in previous work comprise over 1400 different geometries, limiting the applicability of the approach when geometric data is limited and often expensive. Second, GANs rely on a generator-discriminator network architecture that requires meticulous design of both sub-networks to achieve stability. Third, imposing design constraints on the generated samples is cumbersome and often requires training an additional dedicated model.

To address these difficulties, we propose *DiffAirfoil*, a novel approach to sampling that relies on diffusion [13] and operates within an automatically learned latent space of airfoils [14, 15]. As shown in Fig. 1(a), in the unconditional version of our framework, a diffusion network is trained to produce latent vectors that then are fed to the parameterization model to generate an airfoil. In the conditional version depicted by Fig. 1(b), we provide guidance so that the sampled airfoil has user-defined geometric properties. This offers multiple advantages over GANs: First, *DiffAirfoil* is data-efficient and can be effectively trained to sample valid and diverse airfoils with as few as 50 geometries, which is less than 3.5% of the number required by GANs. Second, the training of *DiffAirfoil* is stable and easier to implement, making the model accessible to a broader group of users. Third, *DiffAirfoil* incorporates an automatic parameterization model, ensuring the validity of sampled airfoils, including their contour smoothness and chord alignment, with minimal human effort. Finally differentiable conditions can be directly applied to *DiffAirfoil* as plugins, requiring no additional data or training in implementing the conditional sampling.

In the remainder of this paper, we first describe *DiffAirfoil* in detail. We verify *DiffAirfoil*'s advantages in terms of sampling quality, exploration and condition adherence by performing quantitative and qualitative comparisons with GANs in Section III. We showcase a potential role of the conditional *DiffAirfoil* in the surrogate-based optimization pipeline for 2D airfoil application in Section IV.

II. Method

DiffAirfoil is designed to sample novel airfoils using a shape parameterization model and a function that maps a random normal distribution to the parameterization space. To this end, we rely on the Latent Space Model (LSM) [14, 15] to provide an automatic shape parameterization represented by a learned latent space, denoted as $\mathbf{Z}$. Then we introduce the Latent Space Diffusion Model (LSDM, denoted as g), which progressively denoises a random normal vector into a valid latent code.

A. The Latent Space Model

For *DiffAirfoil*, we employ the LSM to parameterize the airfoil's contour. LSM relies on the auto-decoder design [16, 17] and is trained jointly with the latent space using a dataset of the collected 2D airfoils. This model maps a template airfoil $\hat{M} = \{\hat{V}, E\}$ with given N vertices $\hat{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N\}$ and edges E, such as NACA-0012, conditioned on a d dimensional latent vector $\mathbf{z}$ that parameterizes the output geometry as

$$f_\Theta : \mathbb{R}^d \times \mathbb{R}^2 \to \mathbb{R}^2 \ , \qquad (1)$$
$$\delta\mathbf{v} = f_\Theta(\mathbf{z}, \hat{\mathbf{v}}) \ ,$$

where $\mathbb{R}^2$ stands for the two-dimensional coordinate space. LSM is a multi-layer perceptron (MLP) network and Θ represents the weights that control LSM f. A deformed airfoil contour is obtained as $M = \{\hat{V} + f_\Theta(\mathbf{z}, \hat{V}), E\}$.

During training, we collect a dataset of K airfoils, denoted as $S_1, \ldots, S_K$, which only contains sampled surface points. The training objective is to optimize the weights Θ and the latent vectors that parameterize each training data $\mathbf{z}_1, \ldots, \mathbf{z}_K$, which writes

$$\Theta^*, \mathbf{Z}^* = \underset{\Theta, \mathbf{z}_1, \ldots, \mathbf{z}_K}{\operatorname{argmin}} \sum_{k=1}^{K} \mathcal{L}_{LSM}(\hat{\mathbf{v}} + f_\Theta(\mathbf{z}_k, \hat{\mathbf{v}}), S_k) \ , \qquad (2)$$

where $\mathcal{L}_{LSM}$ is a loss function that is minimized when $f_\Theta(\mathbf{z}_k, \hat{\mathbf{v}})$ yields a deformed airfoil geometrically identical to S_k. The optimal $\mathbf{z}_k^*$ corresponds to a low-dimensional parameterization of S_k. We use the Chamfer Distance [18], denoted as $\mathcal{L}_{CD}(V, S)$, to measure the geometric difference. Additionally, we apply a vector norm regularization on $\mathbf{z}$ to ensure the smoothness of the acquired latent space. Therefore, we have the overall training objective that writes

$$\mathcal{L}_{LSM}(V, S) = \mathcal{L}_{CD}(V, S) + w_\mathbf{z} \, ||\mathbf{z}||^2 \ , \qquad (3)$$

where $w_\mathbf{z}$ is the balancing weight.

At inference time, given a new target geometry S and the frozen weights Θ^*, the latent parameterization vector $\mathbf{z}^*$ is determined by optimizing the following problem:

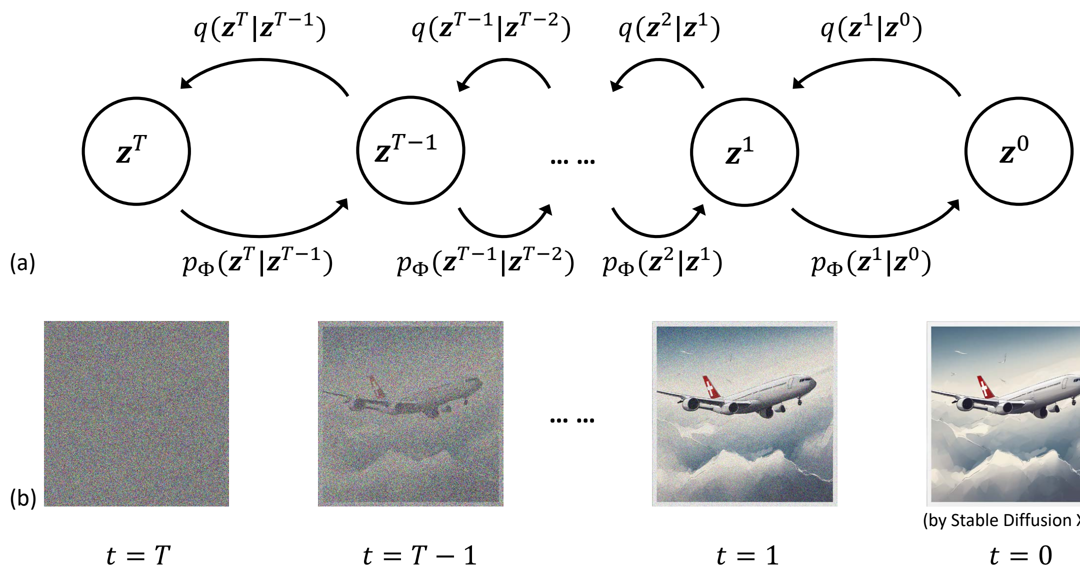
$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \, \mathcal{L}_{LSM}(\hat{\mathbf{v}} + f_\Theta(\mathbf{z}, \hat{\mathbf{v}}), S) \ . \qquad (4)$$

Then the output airfoil contour is obtained by forwarding $\mathbf{z}^*$ with LSM as $M = \{\hat{V} + f_\Theta(\mathbf{z}^*, \hat{V}), E\}$.

To obtain novel geometries instead of encoding a given airfoil, we need to obtain a new latent code within the latent space and then decode it into a deformed airfoil. However, it is intractable to directly sample from the distribution of $\mathbf{Z}$, which is unknown and non-analytical. Therefore, we require the LSDM to sample from a feasible random distribution and project the sampled random vector to a valid latent vector.

B. The Latent-Space-Based Diffusion Model for Unconditional Sampling

The LSDM is a probabilistic generative model that maps a multivariate normal distribution to LSM's latent space by parameterizing a Markov chain. To train the LSDM, we follow the Denoising Diffusion Probabilistic Model (DDPM)



$q(\mathbf{z}^T|\mathbf{z}^{T-1})$
$q(\mathbf{z}^{T-1}|\mathbf{z}^{T-2})$
$q(\mathbf{z}^2|\mathbf{z}^1)$
$q(\mathbf{z}^1|\mathbf{z}^0)$
$\mathbf{z}^T$
$\mathbf{z}^{T-1}$
... ...
$\mathbf{z}^1$
$\mathbf{z}^0$
(a)
$p_\Phi(\mathbf{z}^T|\mathbf{z}^{T-1})$
$p_\Phi(\mathbf{z}^{T-1}|\mathbf{z}^{T-2})$
$p_\Phi(\mathbf{z}^2|\mathbf{z}^1)$
$p_\Phi(\mathbf{z}^1|\mathbf{z}^0)$
(b)
... ...
(by Stable Diffusion XL)
$t = T$
$t = T - 1$
$t = 1$
$t = 0$

Fig. 2 The forward and reverse process of LSDM in (a) directed graph figure. An analogy on an image is shown in (b).

framework [19], defining a forward process, denoted as q, and a reversed process, denoted as p_Φ, as shown in Figure 2(a). An analogous diffusion process on an image is also presented for better clarity in Figure2(b).

The forward process q describes the transitions of the Markov chain that iteratively introduces Gaussian noise into a given latent code $\mathbf{z}$ until the original data is completely destroyed. This step is controlled by the noise variance schedule $\beta_1, \beta_2, ..., \beta_T$ with T discrete steps. The forward process writes

$$q(\mathbf{z}^t|\mathbf{z}^{t-1}) := \mathcal{N}(\mathbf{z}^t : \sqrt{1-\beta_t}\mathbf{z}^{t-1}, \beta_t \mathbf{I}) \ , \tag{5}$$

$$q(\mathbf{z}^{1:T}|\mathbf{z}^0) := \prod_{t=1}^{T} q(\mathbf{z}^t|\mathbf{z}^{t-1}) \ , \tag{6}$$

where $\mathcal{N}$ is the d dimensional Gaussian distribution and $\mathbf{I}$ is the identity matrix. The variance β_t is predefined and remains constant throughout the process. By using the notation $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_t$, the noisy vector $\mathbf{z}^t$ at an arbitrary time step can be efficiently sampled with a closed form and a look up table of $\bar{\alpha}_t$, which writes

$$q(\mathbf{z}^t|\mathbf{z}^0) = \mathcal{N}(\mathbf{z}^t; \sqrt{\bar{\alpha}_t}\mathbf{z}^0, (1-\bar{\alpha}_t)\mathbf{I}) \ . \tag{7}$$

The reverse process of LSDM is the joint distribution $p_\Phi(\mathbf{z}^{0:T})$, also formulated as a Markov chain with learned probability transitions that reduce the previously added noises. It starts at $p(\mathbf{z}^T) = \mathcal{N}(\mathbf{z}^T; \mathbf{0}, \mathbf{I})$, and the entire process can be described as

$$p_\Phi(\mathbf{z}^{0:T}) := p(\mathbf{z}^T) \prod_{t=1}^{T} p_\Phi(\mathbf{z}^{t-1}|\mathbf{z}^t) \ . \tag{8}$$

The reverse transition probability is defined similarly to $q(\mathbf{z}^t|\mathbf{z}^{t-1})$, as $p_\phi(\mathbf{z}^{t-1}|\mathbf{z}^t) := \mathcal{N}(\mathbf{z}^{t-1} : \mu_\phi(\mathbf{z}^t, t), \beta_t \mathbf{I})$. The diffusion network $g_\Phi(\mathbf{z}^t, t)$ of LSDM is a Multi-Layer Perceptron (MLP) network that predicts the noise added in the forward process so that it parameterizes $\mu_\phi(\mathbf{z}^t, t)$ as:

$$\mu_\phi(\mathbf{z}^t, t) := \frac{1}{\sqrt{\alpha_t}}(\mathbf{z}^t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}) g_\Phi(\mathbf{z}^t, t) \ . \tag{9}$$

The model is trained with the Evidence Lower Bound (ELBO) by minimizing the negative log likelihood as:

$$\underset{\Phi}{\operatorname{argmin}}\, \mathbb{E}\left[-\log p_\Phi(\mathbf{z}^0)\right] \leq \underset{\Phi}{\operatorname{argmin}}\, \mathbb{E}_q\left[-\log \frac{p_\Phi(\mathbf{z}^{0:T})}{q(\mathbf{z}^{1:T}|\mathbf{z}^0)}\right] = \mathcal{L}_0 + \sum_{t=2}^{T} \mathcal{L}_{t-1} + \mathcal{L}_T \ , \tag{10}$$

$$\begin{aligned} \text{where } \mathcal{L}_0 &= \mathbb{E}_q\left[-\log p_\Phi(\mathbf{z}^0|\mathbf{z}^1)\right] \ , \\ \mathcal{L}_{t-1} &= \mathbb{E}_q\left[\sum_{t=2}^{T} \mathcal{D}_{KL}\left(q(\mathbf{z}^{t-1}|\mathbf{z}^t, \mathbf{z}^0)||p_\Phi(\mathbf{z}^{t-1}|\mathbf{z}^t)\right)\right], \text{ when } t > 1 \ , \\ \mathcal{L}_T &= \mathbb{E}_q\left[\mathcal{D}_{KL}\left(q(\mathbf{z}^T|\mathbf{z}^0)||p(\mathbf{z}^T)\right)\right] \ . \end{aligned} \tag{11}$$

$\mathcal{D}_{KL}$ is the KL divergence. As β are constants, $\mathcal{L}_T$ also becomes constant. The $\mathcal{L}_0$ and $\mathcal{L}_{t-1}$ are simplified as the loss function of LSDM as

$$\mathcal{L}_{LSDM} := \mathbb{E}_{t,\mathbf{z}^0,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\left\|\epsilon - g_\Phi\left(\sqrt{\bar{\alpha}_t}\mathbf{z}^0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right] \ . \tag{12}$$

The training steps of the diffusion network in LSDM are simple and straightforward, as described in Algorithm 1.

Algorithm 1 The training steps of the diffusion network in LSDM.

```
while not converged do
    sample a latent code from the training dataset as $\mathbf{z}^0$.
    $t \sim \text{Uniform}(\{1, 2, ..., T\})$
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    Take gradient descent optimization step on $\nabla_\Phi \left\|\epsilon - g_\Phi\left(\sqrt{\bar{\alpha}_t}\mathbf{z}^0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2$.
end while
```

According to the score-matching theory [20], the optimized g_Φ is the score function of the reverse process, namely $\nabla_{\mathbf{z}^t} \log p_\Phi(\mathbf{z}^t|\mathbf{z}^{t+1}) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} g_\Phi(\mathbf{z}^t, t)$. Novel latent codes can then be sampled with the stochastic gradient Langevin dynamics [21] from the normal distribution probability density using only the gradients in the Markov chain. This iterative process is described in Algorithm 2.

Algorithm 2 The unconditional sampling steps of LSDM.

```
$\mathbf{z}^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, T-1, ..., 1$ do
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$ , else $\epsilon = 0$
    $\mathbf{z}^{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{z}^t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} g_\Phi(\mathbf{z}^t, t)\right) + \sqrt{\beta_t}\epsilon$
end for
return $\mathbf{z}^0$
```

C. The Conditional Latent-Space-Based Diffusion Model

Generating controllable novel shapes that satisfy the desired geometric properties is essential for many applications. To implement controllable generations, we develop the conditional LSDM (CLSDM). We define the CLSDM as gradient-guided diffusion model, developed from the classifier-guided diffusion model [22] with more generic conditioning.

To formulate CLSDM, let $\hat{C}$ denote the geometric condition, which is the user-defined constant. The airfoil's geometric properties, such as area or maximum thickness, are measurable using differentiable equations, denoted as $c(f_\Theta(\mathbf{z}, \hat{\mathbf{v}}))$, or $c(\mathbf{z})$ for short. The conditioning model can be described from a probabilistic perspective as $p(c(\mathbf{z})|\mathbf{z})$, defined as:

$$p(c(\mathbf{z})|\mathbf{z}, \hat{C}) := \mathcal{N}\left(c(\mathbf{z}); \hat{C}, \sigma_c^2\right); , \tag{13}$$

where σ_c^2 is the standard variation of the distribution of $c(\mathbf{z})$. The idea of CLSDM is to replace the unconditional reverse transition probability $p_\Phi(\mathbf{z}^{t-1}|\mathbf{z}^t)$ with the conditional joint probability $p_\Phi(\mathbf{z}^{t-1}|\mathbf{z}^t, \hat{C}) = p_\Phi(\mathbf{z}^{t-1}|\mathbf{z}^t)p(c(\mathbf{z}^{t-1})|\mathbf{z}^{t-1}; \hat{C})$

during the sampling stage. The new conditional score function $g^c_\Phi(\mathbf{z}^t, t)$ becomes

$$\begin{aligned} g^c_\Phi(\mathbf{z}^t, t) &:= \nabla_{\mathbf{z}^t} \log \left(p_\Phi(\mathbf{z}^t | \mathbf{z}^{t+1}) p(c(\mathbf{z}^t) | \mathbf{z}^t; \hat{C}) \right) \\ &= -\frac{1}{\sqrt{1-\bar{\alpha}_t}} g_\Phi(\mathbf{z}^t, t) + \nabla_{\mathbf{z}^t} \log p(c(\mathbf{z}^t) | \mathbf{z}^t; \hat{C}) \ . \end{aligned} \tag{14}$$

Substitute Equation 13 into Equation 14, we get:

$$g^c_\Phi(\mathbf{z}^t, t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \left(g_\Phi(\mathbf{z}^t, t) + \nabla_{\mathbf{z}^t} \left\| c(\mathbf{z}^t) - \hat{C} \right\|^2 \right) \ . \tag{15}$$

The conditional sampling steps are described in Algorithm 3.

Algorithm 3 The conditional sampling steps of CLSDM.

```
Given: target geometric condition $\hat{C}$
$\mathbf{z}^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, T-1, ..., 1$ do
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$ , else $\epsilon = 0$
    Compute $c(f_\Theta(\mathbf{z}^t, \hat{\mathbf{v}}))$,
    $\mathbf{z}^{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( \mathbf{z}^t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} g_\Phi(\mathbf{z}^t, t) - \nabla_{\mathbf{z}^t} \left\| c(\mathbf{z}^t) - \hat{C} \right\|^2 \right) + \sqrt{\beta_t} \epsilon$
end for
return $\mathbf{z}^0$
```

III. Benchmarking *DiffAirfoil* and Generative Adversarial Networks

Generative Adversarial Networks (GANs) are the state-of-the-art data driven models for sampling novel shapes in aerodynamic applications. In this section, we compare the performance of *DiffAirfoil* and GANs in terms of their training data efficiency, sampling quality, sampling exploration and condition adherence, both qualitatively and quantitatively. *DiffAirfoil* consists of the LSM and the LSDM. The advantages of the automatic LSM have been thoroughly discussed in Wei et al. [14] and Wei et al. [15]. Therefore, we focus on comparing the LSDM with GANs, and the CLSDM with conditional GANs (CGANs), in the same latent space learned by LSM.

A. Generative Adversarial Networks

There are two components in a GAN: a generator and a discriminator. The generator h^G_Γ is a directed latent variable model that deterministically generates samples $\mathbf{z} = h^G_\Gamma(\mathbf{y})$ from random Gaussian vector $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the discriminator h^D_Π is a function that distinguishes between samples from the real dataset $\hat{\mathbf{z}}$ and those generated as $\mathbf{z} = h^G_\Gamma(\mathbf{y})$. The overall optimization problem writes

$$\min_\Gamma \max_\Pi = \mathbb{E}_{\hat{\mathbf{z}} \sim p_{data}} \left[\log h^D_\Pi(\hat{\mathbf{z}}) \right] + \mathbb{E}_{\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\log \left(1 - h^D_\Pi(h^G_\Gamma(\mathbf{y})) \right) \right] \ , \tag{16}$$

CGAN modifyies the generator and the discriminator to take one or more extra scalar input conditions as $\mathbf{z} = h^{CG}_\Gamma(\mathbf{y}, C)$ and $h^{CD}_\Pi(\mathbf{y}, C)$ while using the same training scheme as GAN. During training, the condition is computed from the sampled training airfoil such as $C = \mathbf{c}(f_\Theta(\hat{\mathbf{z}}))$. During inference, C is specified by the user as $C = \hat{C}$.

B. Data Preparation

We collect airfoils from the UIUC database [23], which offers a wide variety of airfoil shapes. Six subsets are randomly sampled, each consisting of 1000, 500, 250, 100, and 50 airfoils, respectively. For simplicity, we use the notion '<*model name*>-<*training data number*>' to represent the model's type and the training setting, such as 'LSDM-1k' and 'GAN-100'. For conditional models, we append another item '<*condition*>', such as 'CLSDM-500-MT12' for airfoil's maximum thickness constrained to 12% chord length or 'CGAN-100-A0.05' for airfoil's area constrained to 0.05.

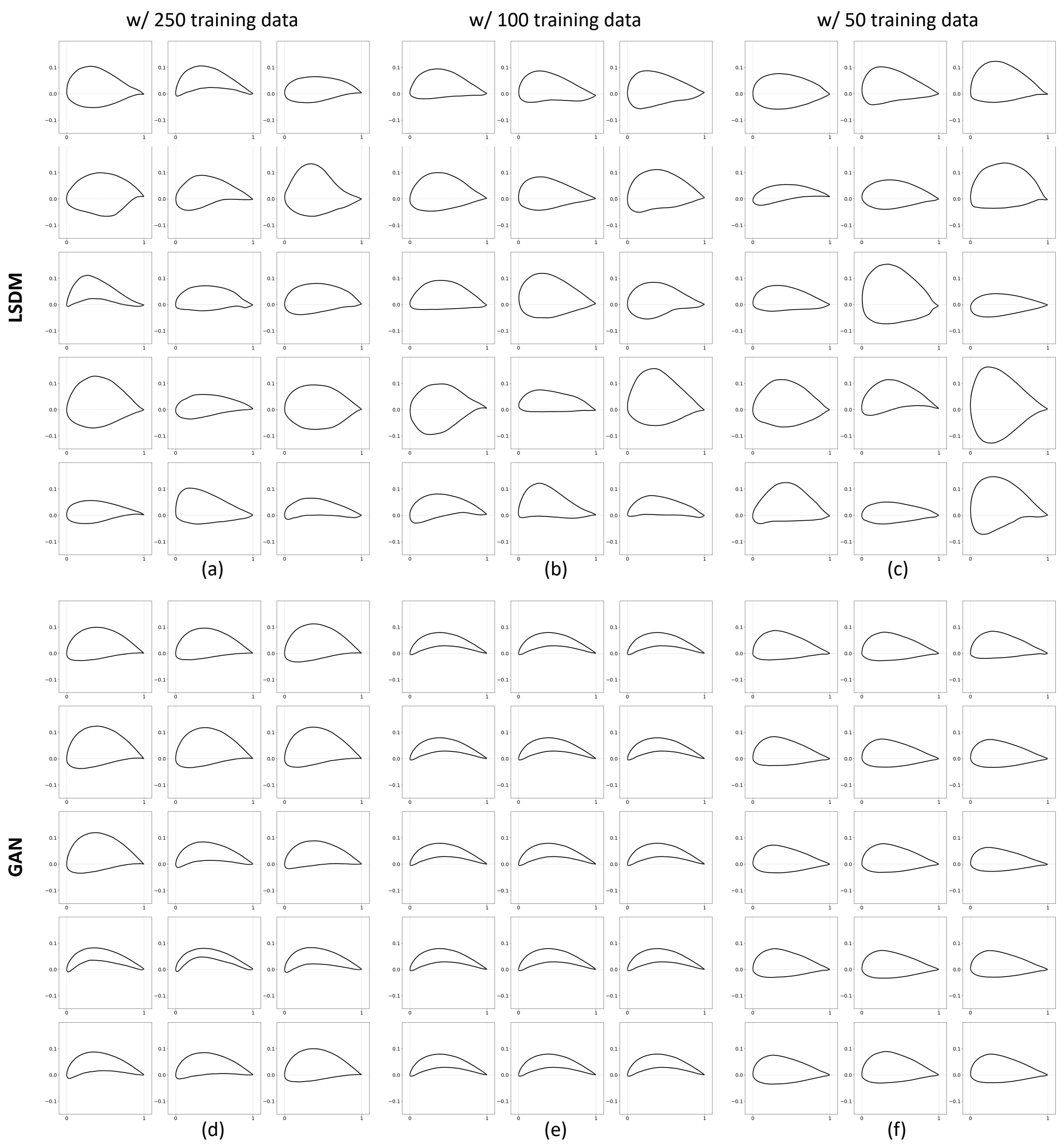


Fig. 3 Airfoils unconditionally sampled by (a) LSDM-250 and (b) GAN-250, (c) LSDM-100 and (d) GAN-100, (e) LSDM-50 and (f) GAN-50.

C. Model Implementation

Both LSM and LSDM are implemented using the PyTorch [*] toolbox and are capable of GPU acceleration. The dimension of LSM's latent space is $d = 256$. We use the Adam optimizer [24] with the learning rate of 5×10^{-4} for LSM's weights Θ and 10^{-3} for the latent space $\mathbf{Z}$.

LSDM is implemented as an MLP model and it has three hidden layers and leaky ReLU activation functions. The

[*]pytorch.org

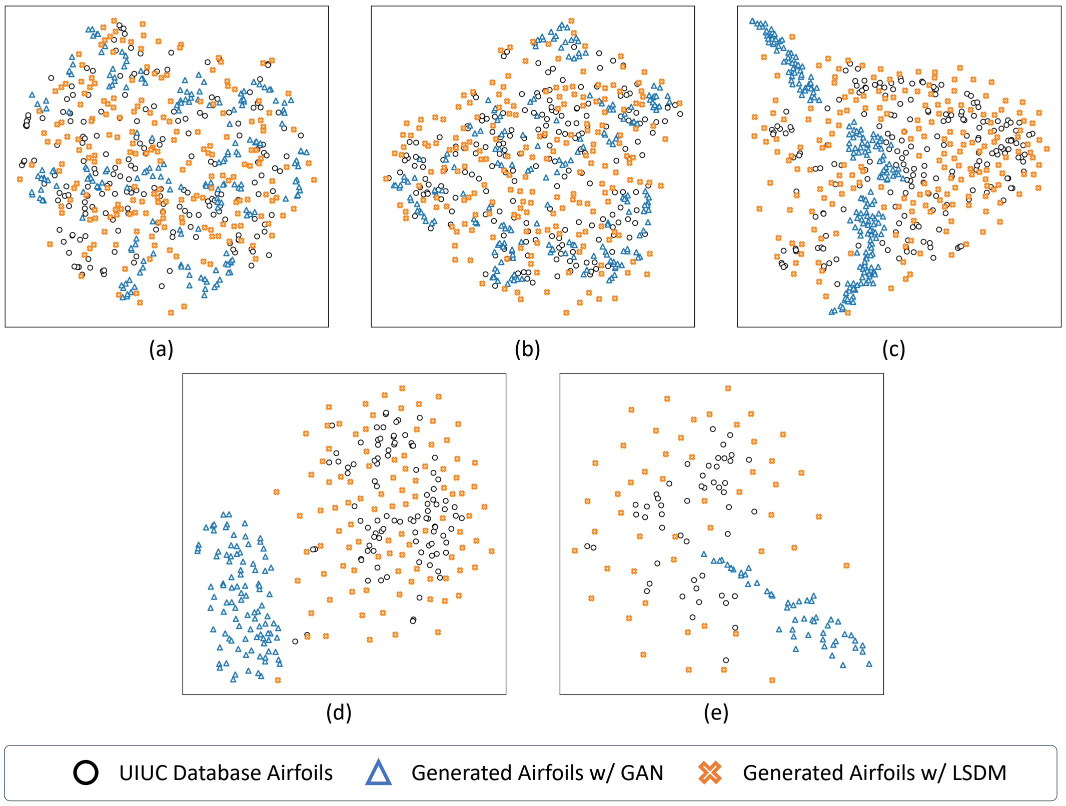


Fig. 4 The *t-sne* visualizations of airfoil's latent codes from the UIUC database, LSDM and GAN, given training datasets of (a) 1000, **(b)** 500, **(c)** 250, **(d)** 100 **and (e)** 50 **airfoils. Best viewed in color.**

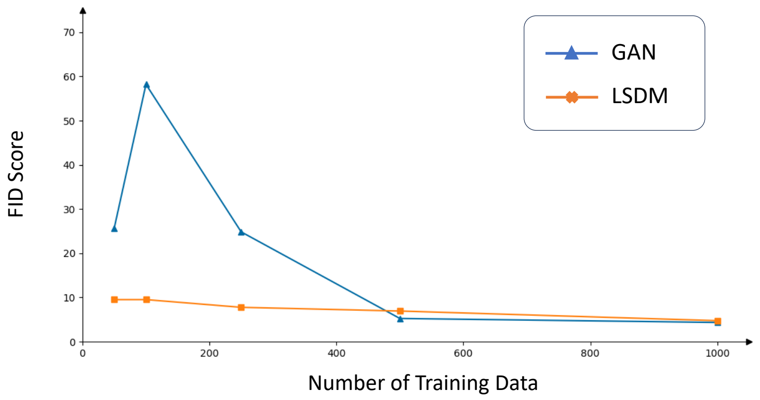


Fig. 5 The FID results of LSDM and GAN when the models are trained with different number of data.

forward and reverse processes use $T = 100$ time steps. The diffusion scheduler β linearly increases from 10^{-4} to 0.02 as t ranges from 0 to T and remains constant in both forward and reverse processes. The LSDM is trained with the AdamW optimizer [25] and a learning rate of 10^{-5}. Under these implementation settings, *DiffAirfoil* is computationally efficient. The training of LSM-50 and LSDM-50 take 671.9 seconds and 716.4 seconds, respectively, on an NVIDIA V100 graphics card. The entire process to sample an novel airfoil takes 61.7 milliseconds.

For GAN and CGAN, both their generator and discriminator networks are implemented as MLP but with two hidden layers. The total number of parameters has to be reduced to ensure both models can be properly trained with the AdamW optimizer and a learning rate at 10^{-5}.

D. Benchmarking Unconditional Generation

This section provides comprehensive comparisons between LSDM and GAN. The study focuses on (i) the quality of generated airfoils, ensuring that the generated airfoils are indistinguishable from the UIUC airfoil dataset, and (ii) on the exploration of generated airfoils, emphasizing the novelty and diversity of new samples.

1. Qualitative Comparison

Figure 3 displays random airfoils sampled by GAN and LSDM. These shapes were generated without any manual filtering, yet they all have valid characteristics: well-aligned chord length, and smooth upper and lower surfaces, which indicates the strong geometric prior embedded in the LSM's latent space. The novel shapes sampled by LSDM show significant variation in thickness, camber, leading-edge radius, and the position of maximum thickness. Notably, LSDM shows no significantly degradation even trained with only 50 airfoils, demonstrating its robustness in handling data scarcity. Meanwhile, the GAN's results reveal severe mode collapses with training data less than 250 where the generation diversity disappears.

In Figure4, we visualize the distribution of latent codes using the t-Distributed Stochastic Neighbor Embedding algorithm (*t-sne*) [26]. The *t-sne* is a statistical dimension reduction method that captures the relative distance information during dimension reduction for visualizing purpose. Therefore, the neighboring points in high-dimensional latent space remain close in the visualized 2D space. Each data point in Figure4 represents an airfoil. In all training settings, LSDM's sampling distributions well aligns with the training airfoils, demonstrating that LSDM has effectively projected the multivariate normal distribution to LSM's latent space. In contrast, GAN's generated airfoil distribution does not align well with the UIUC database, and has bias and limited coverage in the latent space. Severe mode collapses can also be observed, where the generated latent codes show strong clustering effects instead of being separated. These qualitative visualizations clearly illustrate that LSDM has significantly better performance and robustness than GAN under data scarcity.

2. Quantitative Investigation on Sampling Quality

Sampling quality refers to how indistinguishable the generated samples are from the training dataset. Measuring the sampling quality involves solving a distribution-matching problem. We use the Fréchet inception distance (FID) to measure the sampling quality. To implement it, we train a CFD surrogate model on the UIUC airfoil database that predicts the airfoil's lift and drag coefficients following the specifications described in [5]. Given a set of O airfoils, we infer O feature vector for each from the last hidden layer in the surrogate model. Denote the mean feature vector as $\mathbf{M}$ and the covariance matrix as Σ, the FID between two sets of airfoils writes

$$\mathcal{F}(\mathbf{M}_1, \mathbf{M}_2, \Sigma_1, \Sigma_2) = \left[||\mathbf{M}_1 - \mathbf{M}_2||^2 + \mathrm{tr}\left(\Sigma_1 + \Sigma_2 - 2\sqrt{\Sigma_1 \Sigma_2}\right)\right]_F^2 . \tag{17}$$

To quantify the distributional difference between generated airfoils and the UIUC database, we compute M_1 and Σ_1 using the database airfoils. Two FID scores are obtained by computing M_2 and Σ_2 from LSDM's results and GAN's results, respectively.

Figure 5 shows the FID scores. Lower FID value indicates the higher distributional similarity bewteen the generated airfoils and UIUC database. LSDM has a superior overall FID score and its sampling quality remains almost unchanged under severe data scarcity, while GAN totally fails when the training data is less than 500. With sufficient training data, the two models show no significant difference on FID scores.

3. Quantitative Investigation on Sampling Exploration

A generative model that successfully explore the design space will always generate diverse sets of novel designs. We consider two aspects in evaluating the model's sampling exploration: diversity and novelty.

Diversity is a set metric, which measures the generalization ability of the model and the information entropy of generated samples. To implement a diversity measurement, we propose a k-Nearest-Intra-Sample distance, denoted as D^k_{intra}. For each sample in a set of newly generated airfoils, we find its k Nearest Neighborhood (KNN) in the same set. We compute the mean Chamfer Distance (CD) of the KNN with the current sample. Higher CD values indicate more geometric difference between two curves. The overall D^k_{intra} is the averaged mean CDs of all samples. Meanwhile, we also implement a series of statistical diversity measurement on geometric patterns. Given a set of generated airfoils, we compute the standard deviation (STD) of airfoil's area (A), maximum thickness (MT) and camber-to-chord distance (C2C), resulting in three values: STD-A, STD-MT and STD-C2C.

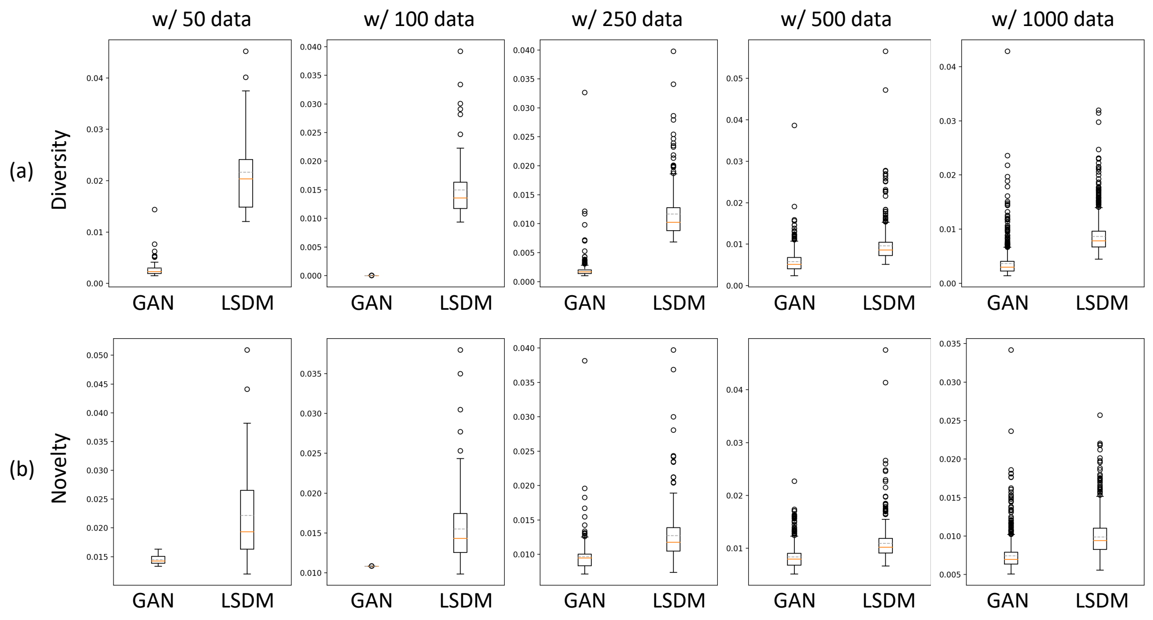


Fig. 6 **The (a) D^{10}_{intra} and (b) D^{10}_{inter} results of LSDM and GAN trained with different number of data.**

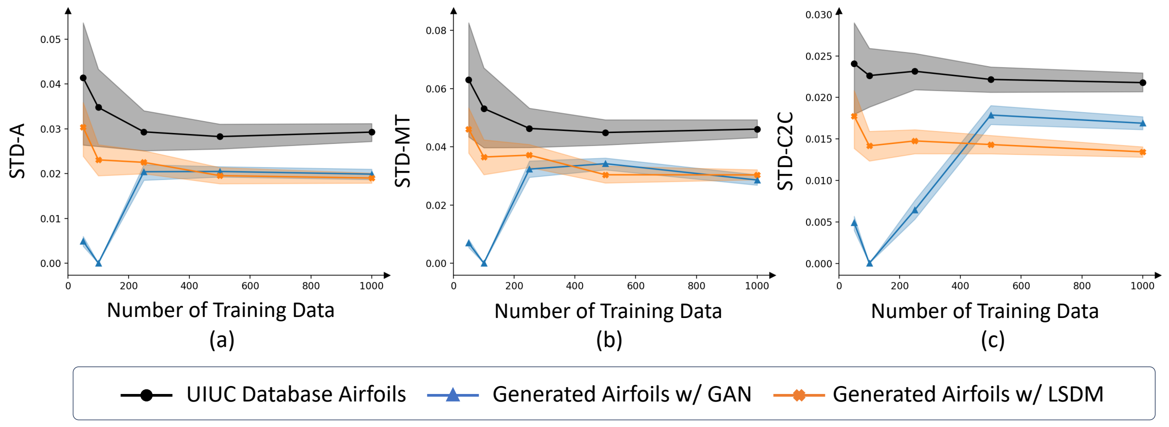


Fig. 7 **The (a) STD-A, (b) STD-MT and (c) STD-C2C results of LSDM and GAN trained with different number of data.**

Novelty is a point metric that measures how unique the generated samples are compared to the training data. We use the k-Nearest-Inter-Sample distance, denoted as D^k_{inter}. For each sampled airfoil, we find its KNN in the UIUC database and compute the mean CD with its KNN. Then D^k_{inter} is the averaged mean CDs of all new samples. For both D^k_{intra} and D^k_{inter}, higher values mean better diversity and novelty performance.

Figure 6 illustrates the D^{10}_{intra} and D^{10}_{inter} results when computed with the 10 nearest neighborhoods. In all settings, LSDM shows significant higher diversity and novelty levels comparing to GAN. With only 100 and 50 training data, GAN shows almost no sampling diversity and novelty, while LSDM exhibits no significant performance drop. Figure 7 demonstrates the STD results of airfoils' geometric patterns. The solid lines are the STD values, while the colored areas represent the 95% confidence range estimated using the bootstrapping sampling. LSDM maintains consistent STD values across all training settings, while severe performance drops can be seen in GAN given less than 200 training data. The difference in STD values corresponds to the previous conclusion that GAN suffers from total failures due to its mode collapse issue.

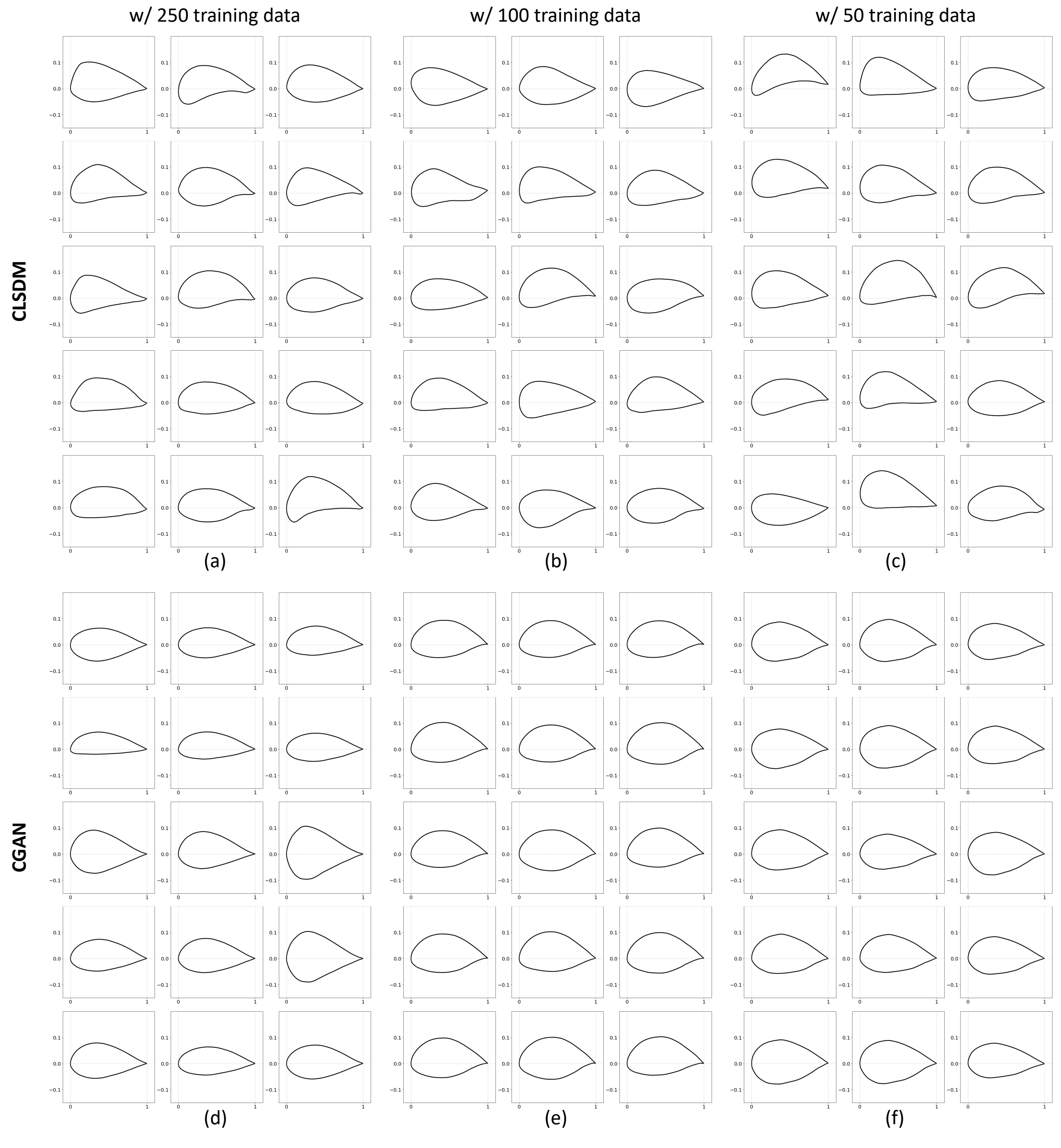


Fig. 8 Airfoils sampled with the condition of area being 0.09 **by (a) CLSDM-250-A0.09 and (b) CGAN-250-A0.09, (c) CLSDM-100-A0.09 and (d) CGAN-100-A0.09, (e) CLSDM-50-A0.09 and (f) CGAN-50-A0.09.**

E. Benchmarking Conditional Generation

This section compares the performance of CLSDM and CGAN. This investigation focuses on the generation satisfaction of the given condition and the model's exploration ability under the condition. To this end, we implement an area condition where we desire the generated airfoil to have a fixed area of 0.09. For CLSDM, we implement the Equation 15 with the area measurement $\mathbf{c}_A$, using the closed-form Shoelace Formula, and $\hat{C} = 0.09$. Assuming the

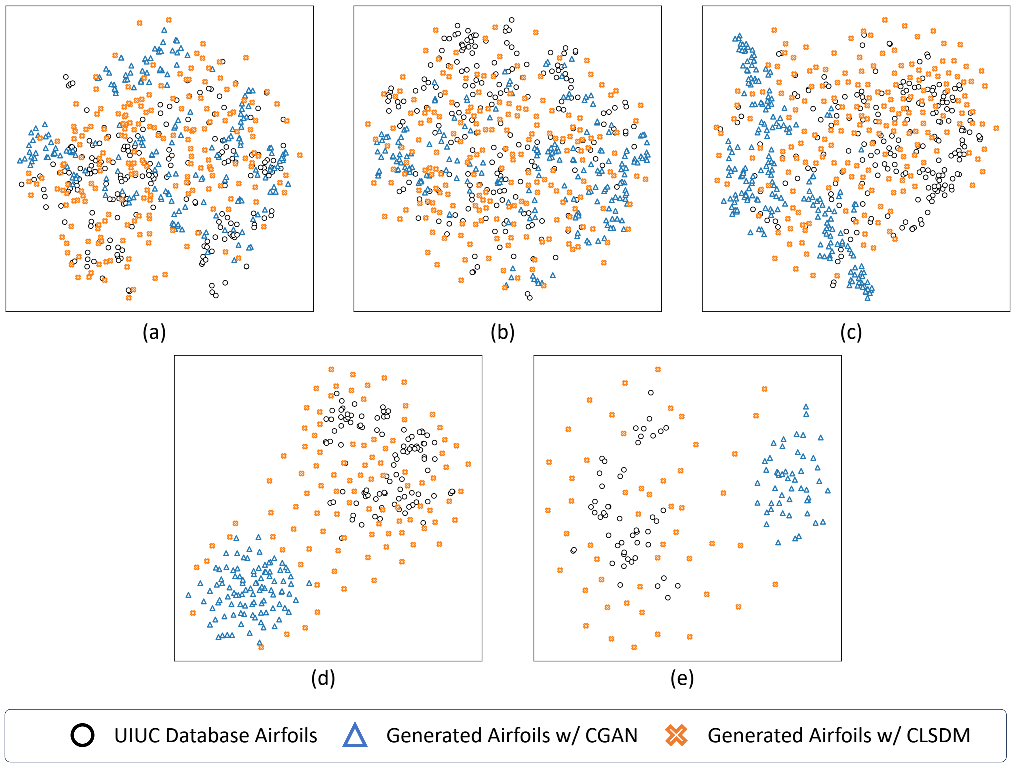


Fig. 9 The *t-sne* visualizations of airfoil's latent codes from the UIUC database, CLSDM and CGAN, given training datasets of (a) 1000**, (b)** 500**, (c)** 250**, (d)** 100 **and (e)** 50 **airfoils, and conditioned on area**= 0.09**. Best viewed in color.**

airfoil vertices $V = \{\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_N\}$ are in clockwise or anti-clockwise order, the $\mathbf{c}_A$ writes

$$\mathbf{c}_A(V) = \frac{1}{2}\sum_{i=1}^{N}\left(v_{i,x}v_{i+1,y} - v_{i+1,x}v_{i,y}\right) \ , \text{ where } \mathbf{v}_i = (v_{i,x}, v_{i,y}) \ . \tag{18}$$

For CGAN, its input condition C is a single scalar. It uses areas of training data for training and a constant $C = 0.09$ for sampling during inference. The CLSDM and CGAN models are denoted as CLSDM-*-A0.09 and CGAN-*-A0.09, respectively.

1. Qualitative Comparison

Figure 8 shows randomly selected airfoils generated by CLSDM and CGAN. All airfoils are valid with normalized chord and smooth curves, thanks to the robustness of LSM. CLSDM demonstrates significantly better diversity than CGAN with the same training data settings, especially given less than 250 training data. Although airfoils sampled by CGAN-250 shows variety, the area condition is not always respected, as demonstrated in Figure 8(d). Figure 9 shows the *t-sne* visualization. Similarly as the conclusion draw from the unconditional generative models, CLSDM aligns better and covers the training data distribution more comprehensively, while the distributions of CGAN are clearly biased with less than 250 training data. Small clusters can been seen in all CGAN's distributions, indicating its poorer generalization ability compared to CLSDM.

2. Quantitative Investigation on Condition Adherence

We study the accuracy of the generated samples in terms of adhering to the desired geometric condition. To this end, we sample airfoils with both CLSDM and CGAN, and compute the absolute difference between the area of sampled airfoils and the condition 0.09. Figure 10 shows that CGAN has significantly higher error than CLSDM, and the error does not decrease with more training data. CLSDM better enforce the user-desired condition during sampling.

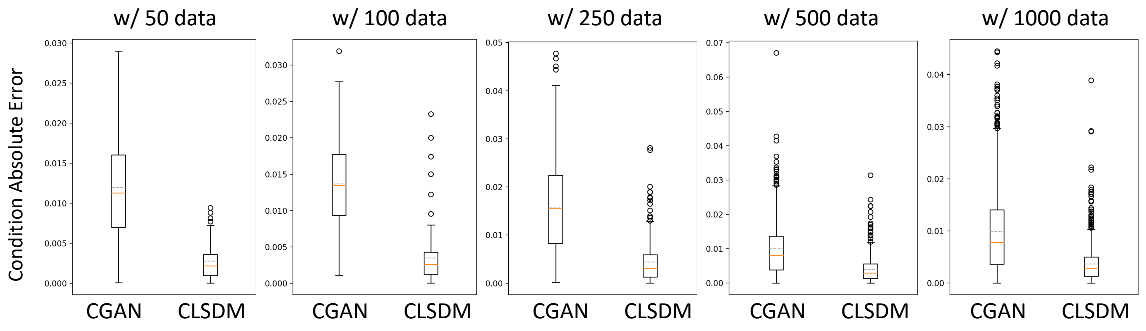


Fig. 10 The mean absolute error between the condition and the area of generated airfoils sampled by CLSDM and CGAN.

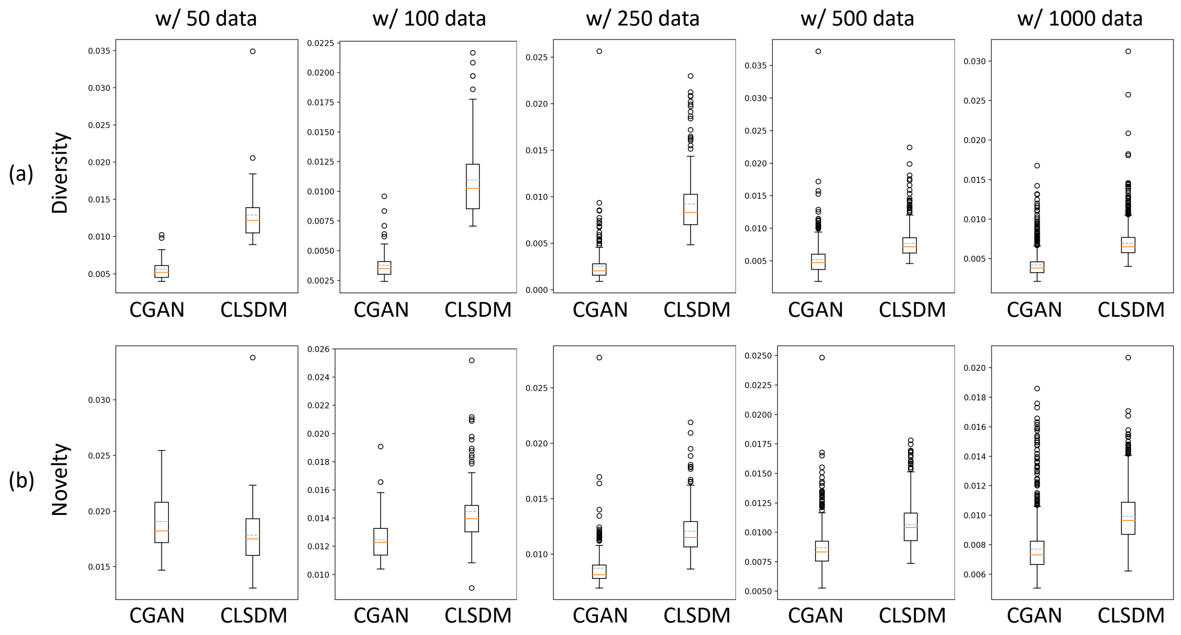


Fig. 11 The (a) D_{intra}^{10} and (b) D_{inter}^{10} results of CLSDM and CGAN trained with different number of data.

3. Quantitative Investigation on Sampling Exploration

We again evaluate the D_{intra}^{10} and D_{inter}^{10} distances to quantify the models' sampling diversity and novelty. Figure 11 presents the results. In general, CLSDM exhibits better sampling exploration than CGAN, similar to the comparison between LSDM and GAN. The difference is less significant, and CGAN appears to be more novel with 50 training data. This is because the CGAN does not adhere well to the area condition, as shown in Figure 10, resulting in greater variation in the sampled shapes. However, this feature is not desirable.

F. Conclusion of Benchmarking

Both qualitative and quantitative investigations conclude that LSDM and CLSDM consistently outperforms GAN and CGAN in terms of the sampling quality and sampling exploration. Under the unconditional setting, all qualitative and quantitative results consistently demonstrate that GANs suffer severe mode collapses given limited training data, with both the sampling quality and exploration heavily degraded. In contrast, LSDM shows almost no impact from data scarcity and exhibits better robustness. For conditional models, CLSDM adheres better to the given condition than CGAN. Additionally, CLSDM is more flexible than CGAN as it requires no retraining or adaptation for different conditions. In conclusion, LSDM and CLSDM are superior alternatives to GAN and CGAN as the data-driven novel shape samplers, especially when training data is limited.

IV. Application to the Surrogate-Based Optimization Pipeline

DiffAirfoil is a useful tool for data generation in the surrogate-based optimization (SBO) pipeline, and we test its ability with a 2D airfoil optimization case in this section.

Given the initial geometry to be optimized, there are two classical approaches to build the CFD dataset for training the surrogate model. The first method is to randomly sample novel geometries by perturbing the initial geometry in a parameterized design space. The second method is to collect similar geometries, e.g., from previous projects, but it is often limited by the availability of data, which may also be irrelevant to the geometry to be optimized.

We demonstrate in this SBO case study that *DiffAirfoil* enables a more effective solution: generate effective geometries for surrogate model training by learning from a limited amount of data with poor relevance to the new optimization task.

A. Optimization Objectives and Experimental Setups

The optimization objective is to maximize the lift-over-drag ratio (L/D) of the NACA-0012 airfoil. The geometric constraint is to fix the airfoil's maximum thickness to 12% chord length. To implement it, we use cubic splines, $spline^{+}(\cdot)$ and $spline^{-}(\cdot)$, to interpolate the surface positions on the upper airfoil and lower airfoil given a horizontal coordinate x, and then calculate the thickness by subtracting the vertical coordinates, which writes

$$\mathbf{c}_{MT} = \max_{x\in(0,1)}\left(\left\|\max(0.12-(spline^{+}(x)-spline^{-}(x)),0)\right\|^{2}\right) \ . \tag{19}$$

The optimization is performed in an inviscid transonic flow with a Mach number at 0.85 and an angle of attack at 0. To run simulations and generate training data, a triangulated mesh built on NACA-0012[†] is used. We use the Direct Mapping Model (DMM) [15] to deform this mesh into different airfoil profiles as the meshing step, and the *SU2*'s Euler equation solver [27] to run simulations.

B. Data Generation

We present the SBO results that use surrogate models with three different training data generation strategies: perturbing random airfoils, collecting limited historical airfoils, and generating airfoils with conditional *DiffAirfoil*, denoted as SM#1, SM#2 and SM#3, respectively.

To train SM#1, we first use a 15th-order Class/Shape Transform (CST) function to fit the NACA-0012 initial airfoil, resulting in the upper curve coefficients $\mathbf{w}_u$, lower curve coefficients $\mathbf{w}_l$ and trailing edge thickness δ_{TE}. Then we sample random coefficients using LHS in different ranges, and the tunning of these ranges has a significant impact on the surrogate model's quality. To illustrate this effect, we develop two variants of SM#1, namely SM#1-1 and SM#1-2. SM#1-1 uses the ranges of $[0.1\mathbf{w}_u, 2.5\mathbf{w}_u]$, $[0.1\mathbf{w}_l, 2.5\mathbf{w}_l]$ and $[-10^{-3}, 10^{-3}]$, respectively. SM#1-2 uses the ranges of $[0.08\mathbf{w}_u, 5\mathbf{w}_u]$, $[0.08\mathbf{w}_l, 5\mathbf{w}_l]$ and $[-10^{-3}, 10^{-3}]$, respectively. 500 airfoils for each model are generated by reconstruction from the randomly sampled coefficients.

To train SM#2, we randomly select 50 shapes from the UIUC database. The sampled shapes include various types of airfoils, hydrofoils and foils for other purposes. Most of the profiles have very different usages from NACA-0012.

To train SM#3, we train the *DiffAirfoil* model using the 50-airfoil dataset for SM#2. Then we follow the conditional sampling steps in Algorithm 3 and implement the guidance in Equation 15 with the maximum thickness measurement $\mathbf{c} = \mathbf{c}_{MT}$ and $\hat{C} = 12\%$. Figure 12 compares the training airfoils for the four surrogate models.

C. Surrogate Model and Optimization Specifications

The surrogate model $sm(V)$ is implemented as an MLP with two hidden layers and ReLU activation function. Its input is a 120-dimensional vector consisting of the x and y coordinates of 60 points from the airfoil's contour. It outputs the lift coefficient C_L and drag coefficient C_D. The loss function is the mean squared error between the predicted and simulated coefficients. The surrogate model is trained using the Adam optimizer for 500 epochs.

The optimization uses LSM and its latent space for parameterization. To perform optimization, we first follow the Equation 4 and infer the latent code of NACA-0012. Then the optimization solves:

$$\mathbf{z}^{*} = \underset{\mathbf{z}}{\operatorname{argmin}}\ ||C_D||^2 - ||C_L||^2 + \mathbf{c}_{MT}(V) + w_{\mathbf{z}}||\mathbf{z}||^2 \ , \tag{20}$$

[†]`https://github.com/su2code/Tutorials/blob/master/design/Inviscid_2D_Unconstrained_NACA0012/mesh_NACA0012_inv.su2`

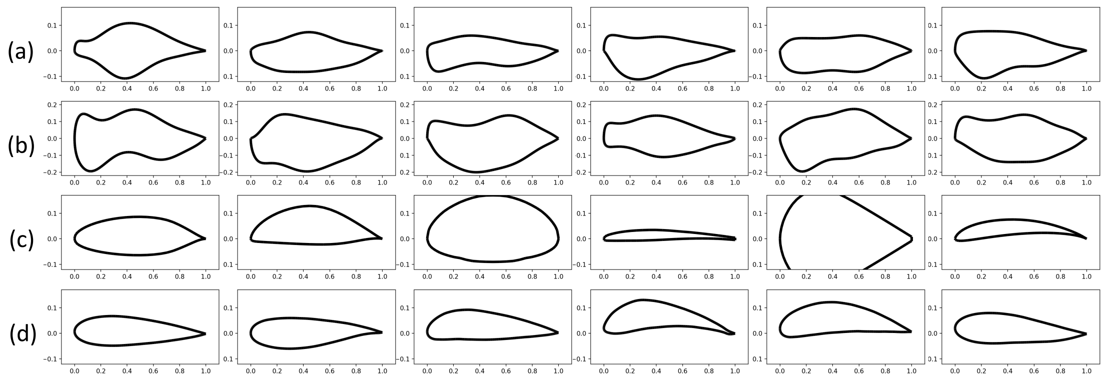


Fig. 12 The airfoils for training (a) SM#1-1, (b) SM#1-2, (c) SM#2 and (d) SM#3. Note that (b) has a different *y* scale.

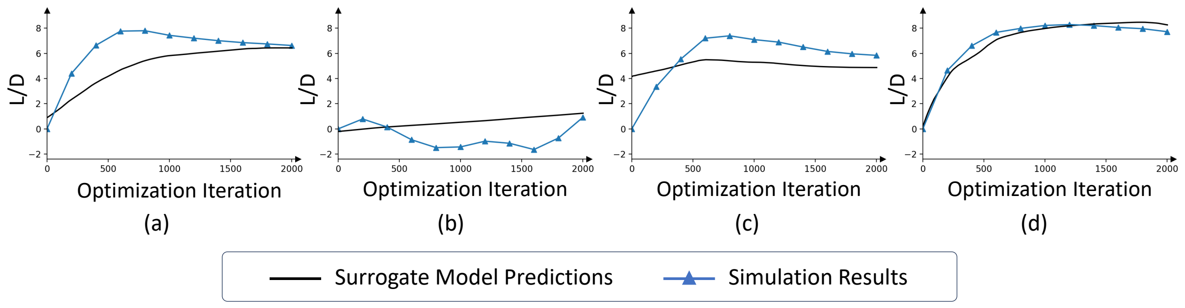


Fig. 13 The change of drag coefficients during the surrogate based optimizations with (a) SM#1-1, (b) SM#1-2, (c) SM#2, and (d) SM#3.

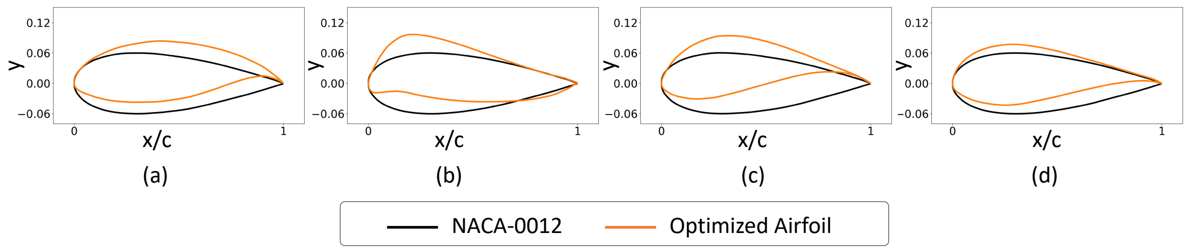


Fig. 14 The initial NACA-0012 and the airfoil optimized by (a) SM#1-1, (b) SM#1-2, (c) SM#2, and (d) SM#3.

where $\{C_D, C_L\} = sm(V)$ and $V = \hat{V} + f_\Theta(\mathbf{z}, \hat{V})$. $w_\mathbf{z}$ is the controlling weight of the regularization of the latent code. The optimization uses Adam optimizer with a learning rate of 10^{-4} for $\mathbf{z}$, and iterates for 2000 times to ensure convergence.

D. Results

The optimization histories are shown in Figure 13. Among the tested models, SM#3 shows the best optimization performance. SM#1-1 and SM#1-2 exhibit significant gap, indicating that parameter tunning of LHS random sampling is crucial to the surrogate model's performance. Additionally, intermediate results were saved for simulation evaluation to compare changes in L/D with the surrogate model's predictions. To quantify prediction performance, we used the R2 score to measure the discrepancy of L/D between simulation results and the surrogate model's predictions. Table 1 presents the quantitative results. A higher R2 score reflects better agreement between the surrogate model and CFD

Table 1 The SBO results and Surrogate Model Accuracy.

Surrogate Model	Final Surrogate L/D	Final Simulation L/D	Surrogate Model Accuracy (R2 Score)
SM#1-1	6.4322	6.6236	0.3265
SM#1-2	1.2432	0.9195	-2.1509
SM#2	4.8718	5.8405	0.2041
SM#3	**8.2603**	**7.7188**	**0.9620**

simulations. SM#3 achieved the highest consistency, indicating it is the most reliable surrogate model. Both SM#1-1 and SM#2 mislead the optimization process, with the best simulation results occurring around the 500th to 600th iterations, but gradually deteriorating afterward. Considering that in real SBO applications the surrogate model is the primary indicator of an airfoil's performance, SM#3 would yield the best optimization results in practice.

The optimized results are compared in Figure 14. The four surrogate models produce different optimized shapes. SM#1-1, SM#1-2 and SM#2 create large cambers, resulting in increased drag and sub-optimal performance. In contrast, SM#3 makes the least shape variation while achieving superior final optimized performance. This suggests that the *DiffAirfoil*-based data sampling strategy provides better coverage of the optimization path, and the surrogate model has better performance in the design space around the initial shape.

V. Conclusion

In this paper, we introduced *DiffAirfoil*, a novel sampling method for generating airfoil shapes using a diffusion model within an automatically learned latent space. Our approach addresses the limitations of traditional sampling methods in high-dimensional design spaces, offering significant improvements in data efficiency and the effectiveness of generated samples.

DiffAirfoil demonstrates several key advantages over conventional Generative Adversarial Networks (GANs). It requires significantly fewer training geometries. The use of standard building blocks and training techniques simplifies its implementation. Moreover, the automatic parameterization method ensures the validity of sampled airfoils, maintaining smooth and well-aligned contours without the need for extensive post-processing. We also highlighted the conditioning capability of *DiffAirfoil*, which allows for the generation of airfoils that meet specific geometric constraints. This feature enhances the practicality of our method in real-world aerodynamic design applications. Our benchmarking results showcase *DiffAirfoil*'s robustness and superior performance compared to GANs, particularly under conditions of data scarcity. By integrating *DiffAirfoil* into the surrogate-based optimization pipeline, we demonstrated its effectiveness in generating highly effective training data, finally leading to improved optimization outcomes.

Future work will explore the extension of *DiffAirfoil* to more complex 3D geometries and further refinement of the conditioning mechanisms to accommodate a wider range of CFD constraints. We believe that *DiffAirfoil* represents a significant step forward in the development of data-driven optimization tools for aerodynamic design, with the potential to streamline and enhance the design process in various applications.

Acknowledgement

Zhen Wei is supported by the French "Programme d'Investissements d'avenir" EUR-TSAE. Michaël Bauerheim is supported by the French Direction Générale de l'Armement through the Agence de l'Innovation de Défense in the project DECAP (Design et Contrôle par Apprentissage Profond) project.

References

[1] Jeong, S., Murayama, M., and Yamamoto, K., "Efficient Optimization Design Method Using Kriging Model," Journal of Aircraft, Vol. 42, No. 2, 2005, pp. 413–420.

[2] Laurenceau, J., Meaux, M., Montagnac, M., and Sagaut, P., "Comparison of Gradient-Based and Gradient-Enhanced Response-Surface-Based Optimizers," American Institute of Aeronautics and Astronautics Journal, Vol. 48, No. 5, 2010, pp. 981–994.

[3] March, A., Willcox, K., and Wang, Q., “Gradient-Based Multifidelity Optimisation for Aircraft Design using Bayesian Model Calibration,” The Aeronautical Journal, Vol. 115, No. 1174, 2011, pp. 729–738.

[4] Toal, D., and Keane, A., “Efficient Multipoint Aerodynamic Design Optimization via Cokriging,” Journal of Aircraft, Vol. 48, No. 5, 2011, pp. 1685–1695.

[5] Baqué, P., Remelli, E., Fleuret, F., and Fua, P., “Geodesic Convolutional Shape Optimization,” International Conference on Machine Learning, 2018.

[6] McKay, M. D., Beckman, R. J., and Conover, W. J., “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code,” Technometrics, Vol. 21, No. 2, 1979, pp. 239–245.

[7] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y., “Generative Adversarial Networks,” Communications of the ACM, Vol. 63, No. 11, 2020, pp. 139–144.

[8] Li, G., Müller, M., Ghanem, B., and Koltun, V., “Training graph neural networks with 1000 layers,” International Conference on Machine Learning, PMLR, 2021, pp. 6437–6449.

[9] Li, J., Zhang, M., Martins, J. R. R. A., and Shu, C., “Efficient Aerodynamic Shape Optimization with Deep-Learning-Based Geometric Filtering,” American Institute of Aeronautics and Astronautics Journal, Vol. 58, No. 10, 2020, pp. 4243–4259.

[10] Du, X., He, P., and Martins, J. R. R. A., “A B-Spline-based Generative Adversarial Network Model for Fast Interactive Airfoil Aerodynamic Optimization,” AIAA Scitech Forum, Orlando, FL, USA, January 6-10, 2020. https://doi.org/10.2514/6.2020-2128.

[11] Chen, W., Chiu, K., and Fuge, M. D., “Airfoil Design Parameterization and Optimization Using Bézier Generative Adversarial Networks,” American Institute of Aeronautics and Astronautics Journal, Vol. 58, No. 11, 2020, pp. 4723–4735.

[12] Achour, G., Sung, W. J., Pinon-Fischer, O. J., and Mavris, D. N., “Development of a Conditional Generative Adversarial Network for Airfoil Shape Optimization,” AIAA Scitech Forum, 2020.

[13] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S., “Deep Unsupervised Learning using Nonequilibrium Thermodynamics,” Proceedings of the 32nd International Conference on Machine Learning, Vol. 37, 2015, pp. 2256–2265.

[14] Wei, Z., Guillard, B., Bauerheim, M., Chapin, V., and Fua, P., “Latent Representation of CFD Meshes and Application to 2D Airfoil Aerodynamics,” American Institute of Aeronautics and Astronautics Journal, 2023. https://doi.org/10.2514/1.J062533.

[15] Wei, Z., Fua, P., and Bauerheim, M., “Automatic Parameterization for Aerodynamic Shape Optimization via Deep Geometric Learning,” AIAA AVIATION Forum, 2023, p. 3471.

[16] Tan, S., and Mayrovouniotis, M. L., “Reducing data dimensionality through optimizing neural network inputs,” AIChE Journal, Vol. 41, No. 6, 1995, pp. 1471–1480. https://doi.org/https://doi.org/10.1002/aic.690410612.

[17] Park, J. J., Florence, P., Straub, J., Newcombe, R. A., and Lovegrove, S., “Deepsdf: Learning Continuous Signed Distance Functions for Shape Representation,” Conference on Computer Vision and Pattern Recognition, 2019.

[18] Barrow, H. G., Tenenbaum, J. M., Bolles, R. C., and Wolf, H. C., “Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching,” International Joint Conference on Artificial Intelligence, 1977.

[19] Ho, J., Jain, A., and Abbeel, P., “Denoising Diffusion Probabilistic Models,” Advances in Neural Information Processing Systems, 2020.

[20] Song, C., Huang, Y., Ouyang, W., and Wang, L., “Box-Driven Class-Wise Region Masking and Filling Rate Guided Loss for Weakly Supervised Semantic Segmentation,” Conference on Computer Vision and Pattern Recognition, 2019.

[21] Welling, M., and Teh, Y. W., “Bayesian Learning via Stochastic Gradient Langevin Dynamics,” International Conference on Machine Learning, 2011, pp. 681–688.

[22] Dhariwal, P., and Nichol, A., “Diffusion Models Beat GANs on Image Synthesis,” Advances in Neural Information Processing Systems, Vol. 34, 2021, pp. 8780–8794.

[23] Selig, M., UIUC airfoil data site, Department of Aeronautical and Astronautical Engineering, University of Illinois at Urbana-Champaign, 1996.

[24] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimisation," International Conference on Learning Representations, 2015.

[25] Loshchilov, I., and Hutter, F., "Decoupled Weight Decay Regularization," International Conference on Learning Representations, 2019.

[26] Maaten, L., and Hinton, G., "Visualizing High Dimensional Data Using t-SNE," Journal of Machine Learning Research, 2008.

[27] Economon, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W., and Alonso, J. J., "SU2: An Open-Source Suite for Multiphysics Simulation and Design," American Institute of Aeronautics and Astronautics Journal, 2016. https://doi.org/10.2514/1.J053813.