DeepGeo: Deep Geometric Mapping for Automated and Effective Parameterization in Aerodynamic Shape Optimization

Zhen Wei *[1], Aobo Yang †[2], Jichao Li ‡[3], Michaël Bauerheim §[4], Rhea P. Liem ¶[2], and Pascal Fua ‖[1]
[1]*EPFL, Lausanne, 1015, Switzerland*
[2]*The Hong Kong University of Science and Technology (HKUST), Hong Kong SAR, China*
[3]*Institute of High Performance Computing, A*STAR, Singapore*
[4]*ISAE-SUPAERO, Toulouse, 31055, France*

Aerodynamic shape optimization (ASO) is a key technique in aerodynamic designs, aimed at enhancing an object's physical performance while adhering to specific constraints. Traditional parameterization methods for ASO often require substantial manual tuning and are only limited to surface deformations. This paper introduces the Deep Geometric Mapping (DeepGeo) model, a fully automatic neural-network-based parameterization method for complex geometries. DeepGeo utilizes its universal approximation capability to provide large shape deformation freedom with global shape smoothness, while achieving effective optimization in high-dimensional design spaces. Additionally, DeepGeo integrates volumetric mesh deformation, simplifying the ASO pipeline. By eliminating the need for extensive datasets and hyperparameter tuning, DeepGeo significantly reduces implementation complexity and cost. Multiple case studies using the same parameterization settings, including the two-dimensional circle-to-airfoil optimization, the three-dimensional CRM wing optimization, and the Blended-Wing-Body aircraft optimization, demonstrate DeepGeo's effectiveness compared to state-of-the-art free-form deformation methods. This research highlights DeepGeo's potential to automate ASO, making it more accessible and efficient.

Nomenclature

α	=	the angle of attack
$C_D,\ C_L,\ C_M$	=	drag, lift and pitching moment coefficients
$f_\Theta,\ \Theta$	=	the Deep Geometric Mapping model and its weights
g	=	the adjoint solver
L	=	the loss function and the soft constraint
M, V, E	=	the CFD mesh and its vertices, edges
O	=	the optimization objective
P	=	FFD control points
S	=	the initial object geometry for optimization
t, A, Vol	=	the geometry's thickness, area and volume
$V^S,\ V^F,\ V^V$	=	CFD mesh vertices on the objects surface, fixed patches and in the volumetric grids

I. Introduction

Shape parameterization is central to Aerodynamic Shape Optimization (ASO) whose objective is to describe and represent an object through variables and mathematical models that implement efficient and effective geometry manipulation. It critically influences the quality of ASO in several ways. Firstly, the parameterization sets the boundaries

*Doctoral Student, Computer Vision Lab, School of Computer and Communication Sciences, AIAA Student Member
†Doctoral Student, Department of Mechanical and Aerospace Engineering, AIAA Student Member
‡Scientist, Fluid Dynamics Department, AIAA Member
§Associate Professor, Department of Aerodynamics, Energies and Propulsion
¶Associate Professor, Department of Mechanical and Aerospace Engineering, AIAA Member
‖Professor, Computer Vision Lab, School of Computer and Communication Sciences

by determining the maximum geometric complexity ASO can handle, the allowable extent of geometric changes it can make, and the surface smoothness it can ensure. Secondly, it must balance between having a large number of parameters that allow for detailed modeling but can limit optimization algorithms' effectiveness, and having fewer parameters that simplify optimization but make accurate representation harder. Thirdly, shape parameterization determines the level of human intervention required, including surface modeling, restricting design variables for different optimizers, and processing the deformation of CFD computational meshes, thus affecting the overall cost of ASO.

Due to its significance, shape parameterization has been extensively studied over the years. For two-dimensional shapes, Hicks-Henne bump functions [1], non-uniform rational B-spline (NURBS) curve and class-shape function transformation (CST) [2] are commonly used, while free-form deformation (FFD) [3–5] is typically implemented for three-dimensional problems. Despite their success across various applications, these methods have notable limitations. Firstly, the quality and computational cost of ASO are highly sensitive to the choice of shape parameterization [6–8], including the selection of algorithms and hyperparameters, with no formal methods to determine the best one. As a result, parameterization design often relies on trial-and-error and empirical priors or adopts methods from similar optimization cases followed by manual tuning. Tradeoffs must be made between deformation freedom, high-dimensional optimization effectiveness, and global shape smoothness. Secondly, existing parameterizations model only the object's surface, requiring an additional module to process the volumetric CFD meshes either through deformation or re-meshing, which introduces further implementation complexities. Additionally, The recent emergence of data-driven parameterization models alleviates some of these issues but demands additional engineering efforts for large-scale geometric datasets in ASO implementation. However, this raises its own challenges because training shapes are often few and expensive to generate in practice.

In this work, we introduce a *Deep Geometric Mapping* (DeepGeo) model to address these difficulties. It uses a deep neural network to represent the target shape and the surrounding CFD computational volumetric mesh required to perform CFD simulations and estimate performance. Joint deformation of both the initial shape and the computational mesh are obtained by changing the network weights. Shape optimization is achieved by minimizing an appropriate objective function with respect to the network weights using an adjoint-based method [9]. Thus, the model's weights can be understood as the CFD mesh parameterization and the computational mesh can be recomputed directly from the parameterization when the target shape changes.

This approach leverages a known strength of neural networks: They can have millions of parameters without suffering from over-parameterization, which helps mitigate the curse of dimensionality that plagues other kinds of models, such as the classical CMAES [10] that is rarely used when there are more than 100 parameters. Moreover, only the target shape and a single template mesh are required to learn an initial set of weights, eliminating the need for a training dataset of meshes and simplifying deployment in practical applications.

In earlier work [11], we introduced an early version of DeepGeo to model 2D airfoils, the Direct Mapping Model. In this paper, we extend it to more challenging optimization tasks and 3D shapes, including 2D circle, 3D wing, and 3D Blended-Wing-Body aircraft optimization.

II. Related Work

DeepGeo spans across multiple research domains, including the shape parameterization algorithms in computer graphics and aerodynamic shape optimization studies, as well as self-prior models in deep learning research. Each part will be briefly reviewed, and the DeepGeo's technical prominence will be discussed afterward.

A. Shape Parameterization in Aerodynamic Shape Optimization

Shape parameterization in ASO refers to the mapping of explicit design variables to the object's surface mesh, a fundamental module for effectively solving ASO problems. Jameson [12] introduced a direct conformal mapping from a circle to optimize airfoils, which provided the maximal design freedom in design space but lacked sufficient smoothness for CFD solvers [13]. Therefore, shape parameterization is necessary to reduce the design space dimensionality and introduce smoothness control. It can be broadly categorized as non-data-driven and data-driven approaches.

1. Non-Data-Driven Approaches

Non-data-driven algorithms include both constructive and deformative methods [6]. Constructive methods, such as polynomial/spline based methods [2, 13, 14] and partial differential equation methods [15, 16], represent the airfoil surface based on specified parameters. Deformative methods deform an existing shape, including analytical

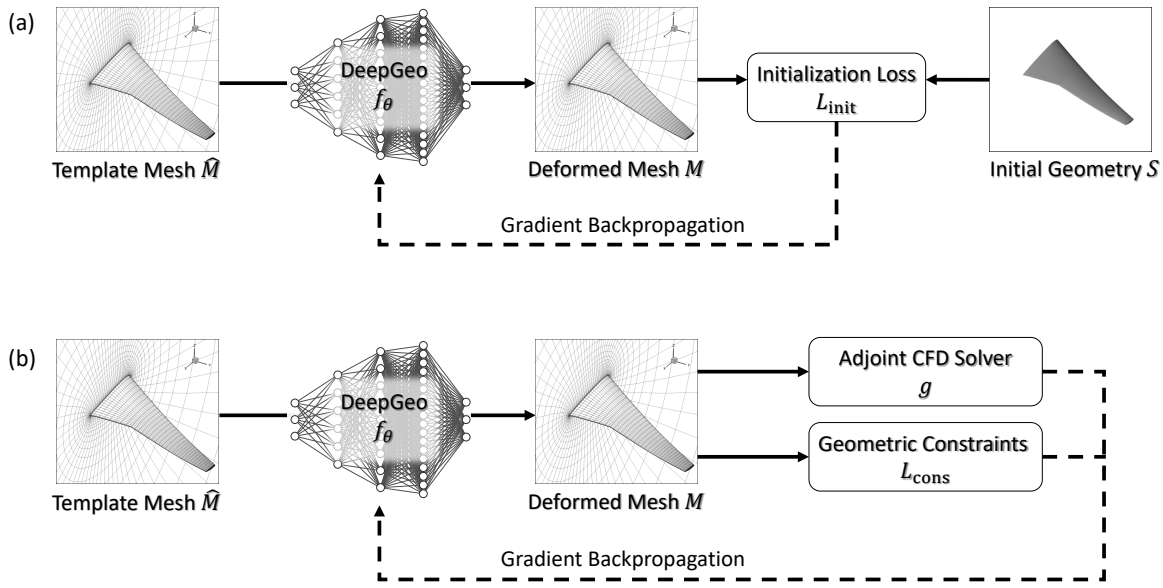

Fig. 1 The ASO pipeline with DeepGeo parameterization. (a) The initialization of DeepGeo learns to deform from template mesh $\hat{M}$ to fit initial geometry S while maintaining mesh quality, and parameterizes S in terms of weights Θ. (b) The shape optimization is performed by varying Θ to minimize an objective function using an adjoint solver and imposing geometric constraints.

methods [1, 17, 18] and the free-form deformation (FFD) [3–5]. Early analytical approaches, like the Hicks-Henne bump function [1] and others [17, 18], optimized the shape through linear combinations of basis functions. B-splines [13] and its variants, like the Bezier curve and nonuniform rational B-spline (NURBS) [14], use control points and piece-wise polynomials to define the geometry. Polynomial-based methods like parameterised sections (PARSEC) [19] and CST [2] approximate surfaces with weighted sums of polynomials with different orders. While effective, these methods are suited for 2D curves. The FFD, originating from soft object animation in computer graphics [3, 4, 20], enables smooth continuous volume transformations based on control point movements. It is widely used in optimizing 3D objects like the CRM wing [21, 22] and the overall aircraft [23].

When applied to infinite-dimensional problems with finite-dimensional design variables, non-data-driven methods require compromising between the dimensionality for effective parameterization and the optimization complexity [24]. Additionally, the lack of self-adaptation leads to heavy reliance on expert knowledge and manual tuning for hyperparamters. As such, different settings can significantly impact optimization results. To address these limitations, DeepGeo bases on the neural network which provides universal approximation ability [25] and mitigates the curse of dimensionality [25, 26].

2. Data-Driven Approaches

Data-driven methods aim to reduce the dimension of design variables using existing geometry datasets, facilitating effective optimization for challenging problems or extracting patterns to reduce human intervention [27]. Linear dimension reduction methods use proper orthogonal decomposition (POD) to derive a set of orthogonal modes. This can be done by Gram-Schmitt orthogonalization [28] or applying a singular value decomposition (SVD) to find orthogonal shape modes from a dataset [29–31]. However, reconstructing the geometry from a latent vector remains challenging and requires dedicated manual designs for specific tasks. The active subspace model (ASM) [32–36] and active subspace identification (ASI) [37] reduce the dimensionality by analyzing the gradient from surrogate models for surrogate-based optimization or uncertainty quantification. Random sampling methods were used to estimate the real active subspace. However, generating valid samples requires handcrafted rules, and the approximation error is upper bounded by the Poincaré constant, which increases with dimensionality given a limited number of samples [38, 39]. Among the non-linear approaches, Viswanath et al. [40] proposed the generative topographic mapping (GTM) that projects a 30-dimensional design variable into a 2D latent vector. GTM's Bayesian generative model makes it challenging to integrate its latent representation into a gradient-based pipeline. More recently, generative adversarial network (GAN)

has been used for novel geometry generation [41–44] to improve the quality of dimension reduction [43, 44] or serve as a parameterization model directly [41, 42].

Data-driven methods typically demand large datasets for training [27]. DeepGeo stands out for its high data-efficiency as it requires only a single initial geometry for training, without requiring any additional user input comparing to the traditional ASO pipeline. This characteristic makes it a feasible solution for complex 3D geometries with limited data availability.

B. Shape Parameterization in Computer Graphics

Shape parameterization in computer graphics aims to create bijective mappings between two surfaces or volumes, with one domain represented as a mesh [45]. This long-standing and active research field has diverse applications, like in mesh morphing, smoothing, remeshing, and so on. Creating one-to-one mappings between coordinate spaces inevitably introduces distortions that need to be minimized to retain desired mesh properties. Research has focused on surface mesh parameterization using cost functions that include distortion metrics, leading to the development of angle-preserving/conformal mappings, area-preserving/authalic mappings [45, 46], as-rigid-as-possible transformation [47, 48], least squares conformal mapping [49] and Dirichlet energy minimization methods [50–52]. DeepGeo extends the elasticity model [52] so that it works with complicated volumetric and unstructured CFD meshes. DeepGeo aligns more closely with the concept of a computer graphics parameterization model, as it establishes a mapping between volumetric CFD meshes instead of generating a design variable vector.

While early ASO shape parameterization methods were inspired by computer graphics, they are not directly applicable to volumetric CFD meshes. DeepGeo's key contribution lies in its regularization loss, which efficiently implements constrained volumetric mesh parameterization. Specifically, DeepGeo's regularization loss is derived from the strain energy of continuous elastic material, eliminating the need for mesh topology and allowing for flexible and highly sparse sampling for loss calculation, thus avoiding unaffordable time and memory consumption on large-scale CFD meshes. Then, DeepGeo uses a neural network, replacing finite difference and finite element approximations with analytical auto-differentiation. Additionally, it uses adjoint gradient descent optimization, which is more effective and computationally efficient than Euler implicit solutions commonly used in computer graphics parameterization models. Third, DeepGeo's approach is independent of mesh topology, which avoids complex mesh processing, especially in cases of extreme cell irregularity where the simple triangulation/tetrahedralization assumption do not hold for CFD meshes.

C. Deep Learning Model with Self-Prior

Self-prior is an emerging research topic in the field of deep learning, where deep neural networks are trained to acquire domain-specific prior knowledge from a single data sample. The deep image prior [53] and its follow-ups have shown strong capabilities in multiple low-level vision tasks with self-supervision, such as image super-resolution, denoising, inpainting, dehazing and deblur. In 3D domain, the deep geometric prior [54] introduced MLP models to reconstruct partial geometry of a point cloud, while Point2Mesh [55] proposed to reconstruct the entire surface mesh from a point cloud.

DeepGeo uses a similar learning technique, but it is the first model that demonstrates how the self-prior can be effectively harnessed and manipulated with the mesh representation under the guidance of an external adjoint CFD solver.

III. Deep Geometric Mapping Model

Optimizing a 3D shape to maximize its performance requires parameterizing it in terms of a set of variables suitable for optimization. In this section, we introduce the Deep Geometric Mapping (DeepGeo) model, which uses its parameters for optimization given only the 3D shape to be optimized. It relies on a neural network f_Θ with weights Θ. To initialize, DeepGeo learns to reconstruct the initial geometry S by deforming from a template CFD mesh $\hat{M}$, embedding the shape parameterization in its weights, as depicted in Figure 1(a). Varying DeepGeo's weights produces variations of the initial shape and properly deforms the corresponding CFD computational mesh, resulting in an output CFD mesh M. During optimization, the deformed CFD mesh is forwarded to an adjoint CFD solver (denoted as g), where the sensitivity of the optimization objectives with respect to the surface vertices is computed. Simultaneously, the gradients of geometric constraints are computed based on the CFD mesh. DeepGeo then backpropagates the surface sensitivity to Θ through auto-differentiation. Thus, the shape can be optimized with respect to Θ, as shown in the pipeline in Fig. 1(b).

This iterative process of gradient calculation and weight update continues until the object reaches an optimum.

A. Formalization

Let $\hat{M} = \{\hat{V}, E\}$ represent a template CFD mesh, where $\hat{V} = \{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, ..., \hat{\mathbf{v}}_N\}$ are the vertices and E denotes its edges. The vertices in $\hat{V}$ can be grouped into three separate sets: $\hat{V}^S$, the vertices on the object surface; $\hat{V}^F$, the vertices from fixed patches if any; and $\hat{V}^V$, the remaining vertices defining the volumetric computational cells. Let $S = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N\}$ be the vertices defining the surface of the initial geometry to be optimized, and S are registered with $\hat{V}$.

We implement DeepGeo as a feed-forward network f_Θ that acts as a continuous mapping between coordinates spaces, defined as:

$$f_\Theta : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \ , \delta\mathbf{v} = f_\Theta(\hat{\mathbf{v}}) \ , \tag{1}$$

where $\delta\mathbf{v}$ is the translation of the input vertex. For brevity, we denote $F_\Theta(\hat{V}) \in \mathbb{R}^{N\times 3}$ as the translation matrix of the N vertices. The deformed mesh becomes $M = \{\hat{V} + F_\Theta(\hat{V}), E\}$.

The initialization of DeepGeo and the shape parameterization step embed the geometric information of the initial shape S into Θ by solving an optimization problem that minimizes the initialization loss function $\mathcal{L}_{init}$ as:

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}}\, \mathcal{L}_{\text{init}}(\Theta, F_\Theta(\hat{V}^S), F_\Theta(\hat{V}^V), S) \ . \tag{2}$$

L_{init} is minimized if and only if the following three conditions are met: (1) the geometric difference between the deformed mesh surface $V^S = \hat{V}^S + F_\Theta(\hat{V}^S)$ and the provided geometry S is minimized, (2) the computational mesh quality is maintained during deformation, and (3) the fixed mesh patches remain unchanged, if any. These conditions are guaranteed by the geometric loss L_{geo}, the mesh regularization loss L_{reg} and the fixation loss L_{fix}, respectively. Thus, L_{init} is written as:

$$\mathcal{L}_{\text{init}} = \mathcal{L}_{\text{geo}}(\Theta) + \lambda_{\text{fix}}\mathcal{L}_{\text{fix}}(\Theta) + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}(\Theta) \ , \tag{3}$$

$$\text{where } \mathcal{L}_{\text{geo}}(\Theta) = \frac{1}{N}\|\hat{V}^S + F_\Theta(\hat{V}^S) - S\|_F^2 \ , \tag{4}$$

$$\mathcal{L}_{\text{fix}}(\Theta) = \frac{1}{\left|\hat{V}^F\right|}\left\|F_\Theta(\hat{V}^F)\right\|_F^2 \ , \tag{5}$$

$$\mathcal{L}_{\text{reg}}(\Theta) = \|\mathbf{H}\left(F_\Theta(V^V)\right)\|_F^2 \ . \tag{6}$$

λ_{reg} controls the strength of $\mathcal{L}_{\text{reg}}$. $\mathbf{H}$ is the Hessian matrix of the network's output with respect to the input vertices. $\mathcal{L}_{\text{geo}}$, $\mathcal{L}_{\text{fix}}$, and $\mathcal{L}_{\text{reg}}$ are explained below.

The Geometric Loss $\mathcal{L}_{\text{geo}}$ is the Euclidean distance between V^S and S.

The Fixed Loss $\mathcal{L}_{\text{fix}}$ measures the magnitude of deformation on fixed patches and constrains their movement, such as on mesh boundaries applied with boundary conditions or on patches constrained by the optimization problem.

The Regularization Loss $\mathcal{L}_{\text{reg}}$ preserves CFD mesh quality by minimizing non-rigid volumetric mesh deformation, avoiding the degradation in cell properties such as skewness, orthogonality and aspect ratio, as well as the mesh's ability to represent the underlying physics [56]. Instead of explicitly constraining any mesh quality measurement, minimizing $\mathcal{L}_{\text{reg}}$ is implemented as a global energy optimization problem. To this end, we treat the coordinate space as a continuous elastic material, and minimizing $\mathcal{L}_{\text{reg}}$ adds resilience against distortion, inducing the deformed cells to resemble those in the template mesh, so as to prevent serious issues such as negative volume and severely non-orthogonality.

To derive $\mathcal{L}_{\text{reg}}$, we define an energy function $\mathbb{E}$ that quantifies the total non-rigid distortion as the squared Frobenius norm of the infinitesimal strain tensor. We have

$$\begin{aligned}\mathbb{E} &:= \frac{1}{2}\left\|\nabla F_\Theta(\hat{V}^V) + \nabla F_\Theta(\hat{V}^V)^T\right\|_F^2 \\ &= \frac{1}{2}\sum_{i=1}^{|V^V|}\left\|\begin{matrix} 2\frac{\partial \delta v_{x,i}^V}{\partial x} & \frac{\partial \delta v_{x,i}^V}{\partial y} + \frac{\partial \delta v_{y,i}^V}{\partial x} & \frac{\partial \delta v_{x,i}^V}{\partial z} + \frac{\partial \delta v_{z,i}^V}{\partial x} \\ \frac{\partial \delta v_{y,i}^V}{\partial x} + \frac{\partial \delta v_{x,i}^V}{\partial y} & 2\frac{\partial \delta v_{y,i}^V}{\partial y} & \frac{\partial \delta v_{y,i}^V}{\partial z} + \frac{\partial \delta v_{z,i}^V}{\partial y} \\ \frac{\partial \delta v_{z,i}^V}{\partial x} + \frac{\partial \delta v_{x,i}^V}{\partial z} & \frac{\partial \delta v_{z,i}^V}{\partial y} + \frac{\partial \delta v_{y,i}^V}{\partial z} & 2\frac{\partial \delta v_{z,i}^V}{\partial z} \end{matrix}\right\|_F^2 \ ,\end{aligned} \tag{7}$$

where $(\delta v^V_{x,i}, \delta v^V_{y,i}, \delta v^V_{z,i}) = \delta \mathbf{v}^V_i$ are the three components of single vertex displacement. An optimal deformed mesh with vertices V^* that has minimal non-rigid distortion should satisfy the Euler-Lagrange equation of $\mathbb{E}$ that stipulates

$$\left\{ 2\frac{\partial^2 v^*_x}{\partial x^2} + 2\frac{\partial^2 v^*_y}{\partial y^2} + 2\frac{\partial^2 v^*_z}{\partial z^2} + \right.$$
$$\frac{\partial^2 v^*_x}{\partial y^2} + \frac{\partial^2 v^*_x}{\partial z^2} + \frac{\partial^2 v^*_y}{\partial x^2} + \frac{\partial^2 v^*_y}{\partial z^2} + \frac{\partial^2 v^*_z}{\partial x^2} + \frac{\partial^2 v^*_z}{\partial y^2} +$$
$$\left. \frac{\partial^2 v^*_x}{\partial xy} + \frac{\partial^2 v^*_y}{\partial xy} + \frac{\partial^2 v^*_x}{\partial xz} + \frac{\partial^2 v^*_z}{\partial xz} + \frac{\partial^2 v^*_y}{\partial yz} + \frac{\partial^2 v^*_z}{\partial yz} \right\} = 0 \ , \forall \mathbf{v}^* \in V^{*V} \ , \tag{8}$$

indicating that V^* sits at a stationary point of the distortion energy system. Equation. 8 is trivially satisfied if all the individual derivative terms are zero. Thus, we define $\mathcal{L}_{reg}$ as

$$\begin{aligned} \mathcal{L}_{\text{reg}} := \sum_{\mathbf{v}^* \in V^{*V}} & \left\| \frac{\partial^2 v^*_x}{\partial x^2} \right\| + \left\| \frac{\partial^2 v^*_y}{\partial y^2} \right\| + \left\| \frac{\partial^2 v^*_z}{\partial z^2} \right\| + \\ & \left\| \frac{\partial^2 v^*_x}{\partial y^2} \right\| + \left\| \frac{\partial^2 v^*_x}{\partial z^2} \right\| + \left\| \frac{\partial^2 v^*_y}{\partial x^2} \right\| + \left\| \frac{\partial^2 v^*_y}{\partial z^2} \right\| + \left\| \frac{\partial^2 v^*_z}{\partial x^2} \right\| + \left\| \frac{\partial^2 v^*_z}{\partial y^2} \right\| + \\ & \left\| \frac{\partial^2 v^*_x}{\partial xy} \right\| + \left\| \frac{\partial^2 v^*_y}{\partial xy} \right\| + \left\| \frac{\partial^2 v^*_x}{\partial xz} \right\| + \left\| \frac{\partial^2 v^*_z}{\partial xz} \right\| + \left\| \frac{\partial^2 v^*_y}{\partial yz} \right\| + \left\| \frac{\partial^2 v^*_z}{\partial yz} \right\| \\ = & \left\| \mathbf{H} \left(F_\Theta(\hat{V}^V) \right) \right\|_F^2 \ , \end{aligned} \tag{9}$$

where $\mathbf{H}$ is the Hessian of f_Θ. Minimizing $\mathcal{L}_{\text{reg}}$ favors fields of deformation vectors with minimal non-rigid distortion. Equation 9 can be efficiently computed via auto-differentiation through DeepGeo's neural network backbone, such as with the implementation of PyTorch *.

B. Implementation Details

DeepGeo is implemented as a multi-layer perceptron (MLP) model. Both the input and output dimensions are three. For 2D ASO cases, the z coordinate is simply set to 0. The default configuration of DeepGeo for all case studies includes three hidden layers, with dimensions set to 64, 256, and 512 from the input side to the output side, respectively. All layers include bias terms and use the layer normalization [57], resulting in a total of 151 686 parameters. The activation function is a sum of ReLU [58] and sine functions, which ensures fast convergence and enables the computation of higher-order derivatives for L_{reg}. DeepGeo's specifications remain fixed for all ASO pipeline for all cases we studied, thus eliminating the need for tuning.

IV. Adjoint-Based Shape Optimization with Deep Geometric Mapping Model

Given the DeepGeo model described above, all ASO case studies discussed in this paper utilize the finite-volume CFD solver ADflow [59] for Reynolds-Averaged Navier-Stokes (RANS) simulation with the Spalart–Allmaras (SA) turbulence model, based on the generated CFD mesh M from DeepGeo. Sensitivities of optimization objectives with respect to the object surface V^S are obtained with ADflow's discrete adjoint solver. Meanwhile, the deformed object surface is evaluated by the geometric constraints, denoted as L_{cons}. These constraints are implemented as differentiable soft constraints, and their gradients with respect to V^S are calculated via auto-differentiation. The gradients of Θ are then computed following the chain rule that begins with the ADflow's surface sensitivities and the gradients of geometric constraints. Θ is updated with the RAdam optimizer.

In short, the overall objective of ASO is:

$$O = O_{\text{CFD}} + w_{\text{cons}} \mathcal{L}_{\text{cons}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} + \lambda_{\text{fix}} \mathcal{L}_{\text{fix}} \ , \tag{10}$$

where w_{cons} is the balancing weights, which depends on the number, dimension and unit of different geometric constraints. The regularization loss $\mathcal{L}_{\text{reg}}$ and the fixed loss $\mathcal{L}_{\text{fix}}$ are as defined in Eq. 3, and the loss weights remain the same. These

*pytorch.org

losses are necessary to maintain CFD mesh quality and freeze the fixed patches during optimization. O_{CFD} represents physical objectives such as drag minimization, lift maximization, etc. It is defined precisely, along with L_{cons}, in each case study below. ASO then involves finding the solution to:

$$\begin{aligned}\Theta^* &= \underset{\Theta}{\operatorname{argmin}} \left(O\left(F_\Theta(\hat{V}), \Theta\right)\right) \\ &= \underset{\Theta}{\operatorname{argmin}} \left(O_{CFD}\left(\left\{\hat{V} + F_\Theta(\hat{V}), E\right\}\right) + w_{cons}\mathcal{L}_{cons}\left(\Theta\right) + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}\left(\Theta\right) + \lambda_{\text{fix}}\mathcal{L}_{\text{fix}}\left(\Theta\right)\right) ,\end{aligned} \tag{11}$$

where $\left\{\hat{V} + F_\Theta(\hat{V})\right\}$ denotes the set of vertices displaced by the translations predicted by f_Θ. This iterative process is stopped based on the early stopping strategy commonly used in gradient-descent-based neural network training [60]. This occurs when O falls below a user-defined threshold or shows no significant improvement after a certain number of iterations.

DeepGeo's specifications remain fixed when implementing the ASO pipeline for different cases, making it a tuning-free and automatic parameterization module. In the following sections, the effectiveness of the DeepGeo-based ASO framework is verified through three different case studies, including tasks that optimize a 2D circle, a 3D business jet wing, and a 3D Blended-Wing-Body aircraft.

V. Case Study I: Optimization from Circle

This section presents a 2D case study where the objective is to minimize the drag from a circle, with the optimization expected to converge to an airfoil. This case involves significant geometric manipulation and changes in the fluid field, testing the stability of DeepGeo's parameterization for maintaining global shape smoothness and its mesh deformation ability to handle large shape variations.

A. Problem Formulation

The initial geometry is a 2D circle with a diameter of 1. The objective is to minimize the drag coefficient C_D while maintaining the lift coefficient C_L at 0.824 and constraining the pitching moment coefficient to $C_M \geq -0.092$. The optimization is conducted in a transonic turbulent flow with a Mach number of 0.734, a Reynolds number of 5×10^6 and an initial angle of attack of 2.0°. The initial mesh, built on the circle using pyHyp [23], consists of 20 331 cells, as shown in Figure 2(a).

Table 1 Aerodynamic shape optimization task specifications for the Case Study I.

Objectives	Functions/Variables	Description	DeepGeo Quantity	FFD Quantity
Minimize	C_D	Drag coefficient		
With respect to	Θ	Weights of DeepGeo	151 585	
	P_z	Control points' z coordinates		30
	α	Angle of attack	1	1
Total design variables			151 586	31
Subject to	$C_L = 0.824$	Lift coefficient constraint	1	1
	$C_M \leq 0.092$	Moment coefficient constraint	1	1
	$t \geq 0.2\, t_{\text{RAE2822}}$	Minimum thickness constraints		30
	$A \geq 0.0654$	Minimum area constraints	1	1
	$\delta\mathbf{v}^{\text{TE}} = 0$	Trailing edge constraint	1	
	$\delta\mathbf{v}^{\text{LE}} = 0$	Leading edge constraint	1	
	$0.5 \leq \alpha \leq 4.0$	Angle of attack constraint	1	1
Total constraints			6	34
Need value range limits for each DV?			NO	YES

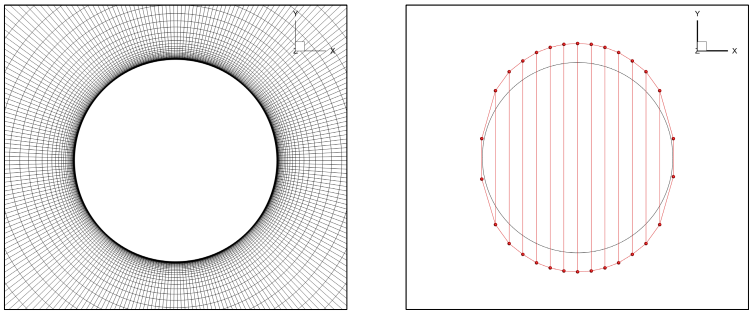

Fig. 2 The geometric setups for for Case Study I, including: (a) template mesh for DeepGeo and (b) the control point setting for FFD baseline.

B. Configuration of Deep Geometric Mapping Model

Following the strategy designed by Li and Zhang [44], a two-step optimization is performed. First, the drag coefficient C_D is minimized. Once the optimization converges, constraints for the lift coefficient C_L and the pitching moment coefficient C_M are added, and the optimization continues as the second stage. The two objectives, $O_{CFD,1}$ and $O_{CFD,2}$, are defined as

$$\begin{aligned} O_{CFD,1} &:= |C_D| \ , \\ O_{CFD,2} &:= |C_D| + |C_L - 0.824| + \max\left(-0.092 - C_M, 0\right) \ , \end{aligned} \tag{12}$$

where the coefficients are obtained with the adjoint solver as $(C_D, C_L, C_M) = g\left(\{\hat{V} + F_\Theta(\hat{V}), E\}\right)$. The airfoil's area, denoted as $A(V^S)$, is calculated from the deformed surface using the Shoelace formula. The geometric constraint consists of a terms that avoid the deformed area to be less than the area of RAE 2822 profile (namely 0.0654), and two

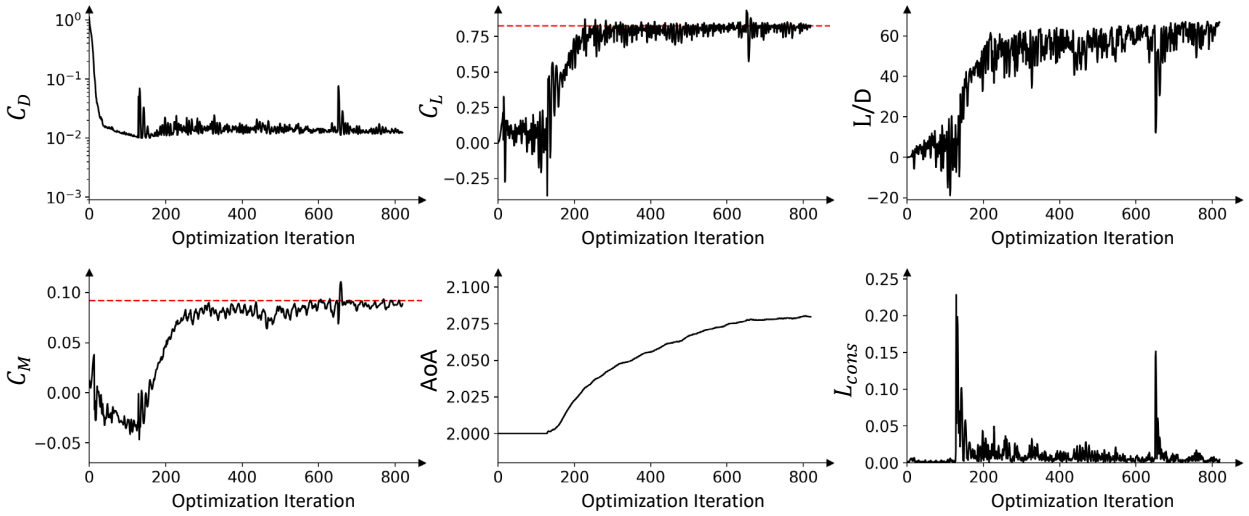

Fig. 3 The optimization history of Case Study I. The dashed line in C_L and C_M figures demonstrates the optimization objectives.

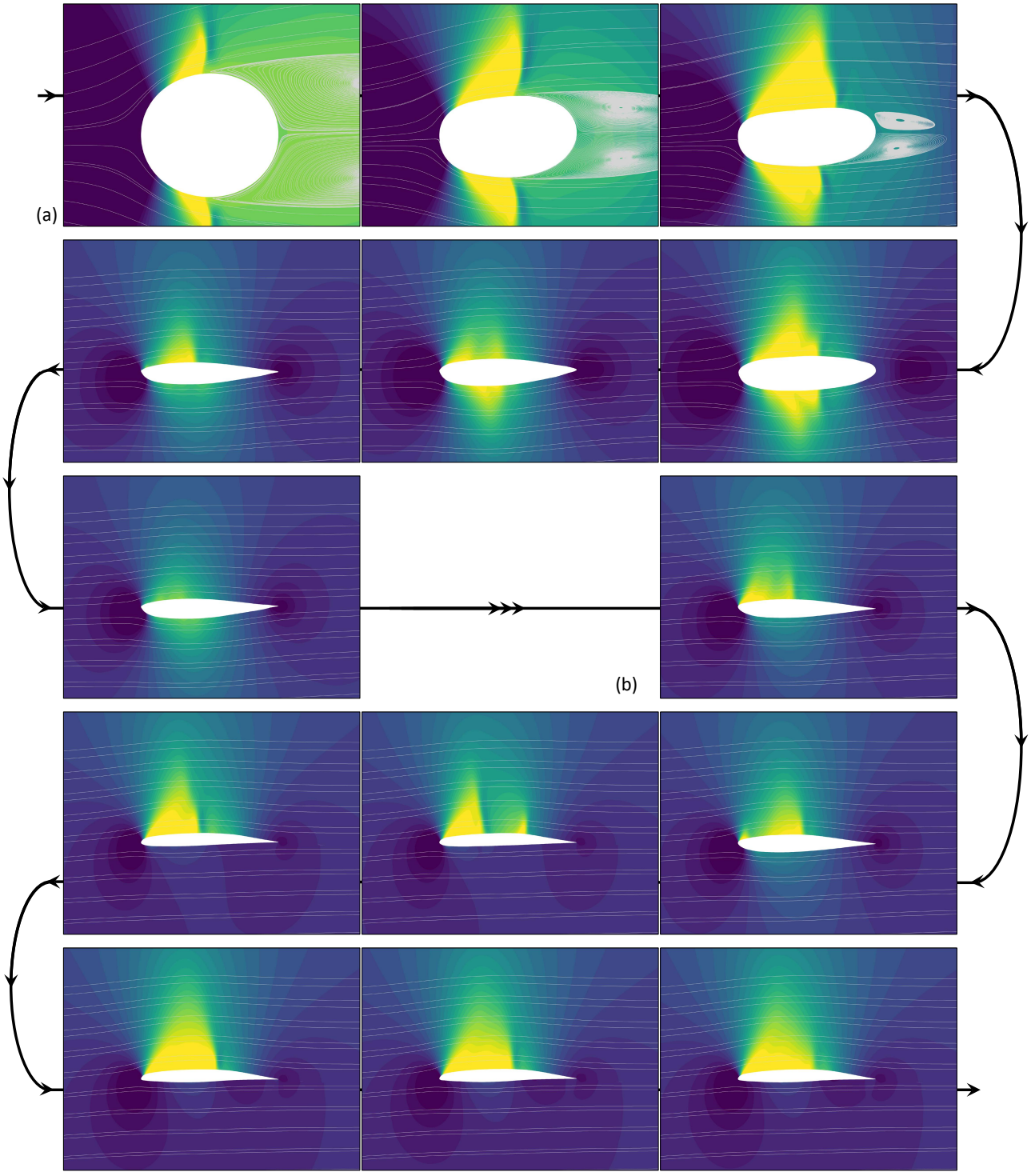

Fig. 4 **The change of pressure coefficient field in Case Study I during the optimization with DeepGeo, including (a) the first optimization stage and (b) the second optimization stage.**

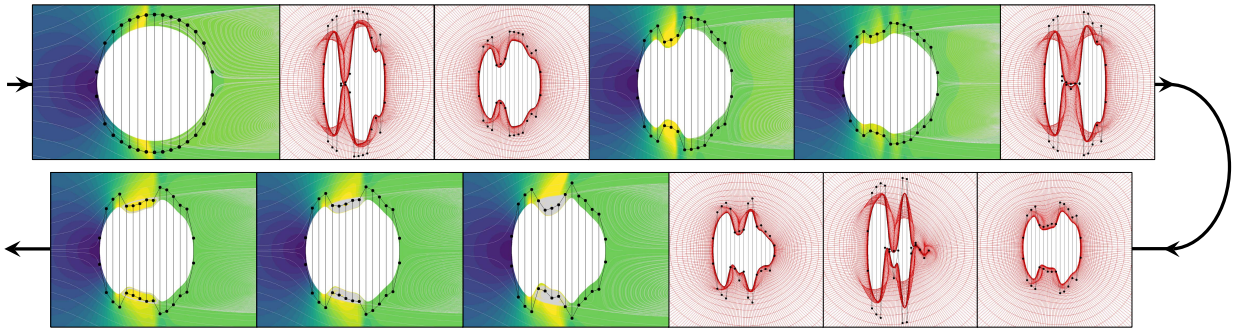

Fig. 5 The FFD-based optimization history in Case Study I. The absence of fluid field means a failed simulation.

terms that fix the LE and TE vertices, which is defined as:

$$\mathcal{L}_{cons} = \underbrace{\max\left(0.0654 - A(V^S), 0\right)^2}_{\text{area constraint}} + \underbrace{\left\|\delta\mathbf{v}^{\text{LE}}\right\|^2}_{\text{LE constraint}} + \underbrace{\left\|\delta\mathbf{v}^{\text{TE}}\right\|^2}_{\text{TE constraint}} . \tag{13}$$

A summary of the optimization setups is shown in Table 1.

DeepGeo uses the initial mesh as its template mesh. During parameterization, DeepGeo learns to output an all-zero deformation since the template mesh is built exactly on the initial mesh, a process we call *self-initialization*. Empirical evidence confirms that self-initialization does not lead to an all-zero trivial solution for Θ and does not degrade the expressiveness of DeepGeo.

C. Configuration of Free-Form Deformation Baseline

For baseline comparison, we use the study case implemented in [44] with a 30-point configuration for FFD, as shown in Figure 2(b). Denote the airfoil's thickness as t, and t_{RAE2822} as the thickness of the RAE 2822 airfoil. A set of thickness constraints $t \geq 0.2\, t_{\text{RAE2822}}$ is applied to avoid the negative volume error at the beginning of optimization. The SLSQP optimizer in pyOptSparse [61] is used for optimization.

D. Results and Analysis

The optimization history and the visualization of fluid field changes are demonstrated in Figure 3 and Figure 4, respectively. In the first optimization step with DeepGeo, the drag coefficient C_D drops significantly. The shape gradually and smoothly becomes thinner, with symmetry mostly maintained. The leading and trailing edges begin to appear. No singular shapes or simulation failures occur during this process, indicating the strong robustness of parameterization and mesh deformation. In the second stage, the shape's symmetry is intentionally broken to achieve a streamline commonly seen in high-lift transonic airfoils, so as to generate more lift. A shock wave initially appears and is then gradually diffused by the end of optimization. The final airfoil achieves $C_D = 0.012\,195$ (i.e. 121.95 counts), and its lift-over-drag ratio (L/D) is improved from 1×10^{-3} to 67.22, comparable to modern supercritical airfoil well-developed for transonic flights.

As a comparison, the FFD-based optimization failed after a small number of iterations due to the singularity of the deformed shape and subsequent severe meshing issues. The optimization process shown in Figure 5 contains discrete shapes that fail in CFD simulations. The iterations visualized only with the mesh mean failures in the simulation. Although FFD ensures smoothness within a local control cage using spline interpolation, the control points as design variables are local and can lead to abnormalities from a more global perspective. The failure of FFD-based optimization persists regardless of the selection of hyperparameters. DeepGeo outperforms FFD in terms of providing global smoothness while still enabling sufficient freedom on the surface and stability in the volumetric mesh deformation.

VI. Case Study II: the Common Research Model Wing Optimization

This section presents the DeepGeo-based ASO results for NASA's Common Research Model Wing (CRM wing) and compares them with the well-established FFD-based ASO results.

A. Problem Formulation

The initial geometry is a three-dimensional wing-only geometry with a blunt trailing edge, extracted from the CRM wing-body configuration, as shown in Figure 6(b). The objective is to minimize its drag coefficient C_D with the lift coefficient C_L constrained to 0.5, and the pitching moment coefficient constrained to $C_M \geq -0.17$. This single-point optimization problem is performed under fully turbulent flow conditions at a Mach number of 0.85, Reynolds number of 5×10^6 and an initial angle of attack of 2.2°. The optimization is conducted on the L2 grid, as shown in Figure 2(a), which has 450 560 cells. For geometric constraints, the wing's volume must be greater than or equal to its initial value, and the thickness must be at least 25% of the initial thickness. All trailing edge (TE) vertices and the leading edge (LE) vertex on the root section are fixed. The other LE vertices are only allowed to move along the the z direction. The projected area is fixed if the constraints on LE and TE are satisfied. The parameterization manipulates the wing's surface, and the angle of attack is treated as a standalone design variable. Table 2 summarizes the optimization problem.

B. Configurations of Deep Geometric Mapping Model

DeepGeo uses the L2 grid as the input template mesh. DeepGeo's physical objectives are defined according to the optimization problem statement as:

$$O_{\text{CFD}} = |C_D| + |C_L - 0.5| + \max\left(-0.17 - C_M, 0\right) \ , \tag{14}$$

Meanwhile, the constraints write:

$$\mathcal{L}_{cons} = \underbrace{\max\left(Vol(V^S) - Vol_{\text{original}}(S), 0\right)^2}_{\text{volume constraint}} + \underbrace{\left\|\Delta V^{\text{TE}}\right\|^2}_{\text{TE constraint}} + \underbrace{\left\|\Delta V_x^{\text{LE}}\right\|^2 + \left\|\Delta V_y^{\text{LE}}\right\|^2}_{\text{LE constraint}} + \underbrace{\left\|\delta \mathbf{v}^{\text{LE,root}}\right\|^2}_{\substack{\text{fixed-wing root} \\ \text{incidence constraint}}} \ , \tag{15}$$

where the volume $Vol(V^S)$ of the deformed wing is computed using the extended Shoelace formula [62] with an arbitrary reference point on the root section plane, and $Vol_{\text{original}}(S)$ is the initial wing volume. $\mathbf{v}^{\text{LE,root}}$ is the single LE vertex on the root section. V_x and V_y refer to the sets of x and y vertex coordinates, respectively. DeepGeo can satisfy the thickness requirement without an explicit constraint.

C. Configurations of Free-Form Deformation Baseline

The FFD-based baseline optimization follows the 192-point setting provided by Hwang et al. [63] and Li et al. [30], as shown in Figure 6(c). The FFD control points are denoted as $P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{192}\}$, where $\mathbf{p} = \{p_x, p_y, p_z\}$, and

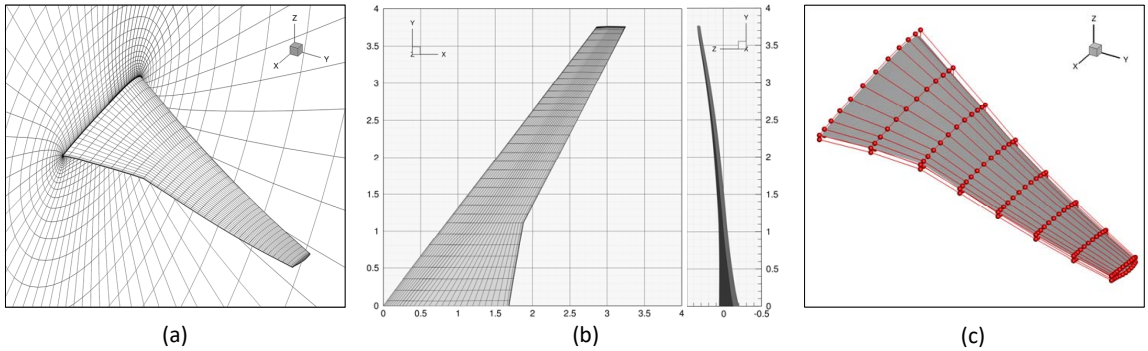

Fig. 6 The geometric setups for Case Study II, including: (a) the template mesh for DeepGeo, (b) the initial CRM wing geometry and (c) the 192-point FFD setting.

Table 2 Aerodynamic shape optimization task specifications for the Case Study II.

Objectives	Functions/Variables	Description	DeepGeo Quantity	FFD Quantity
Minimize	C_D	Drag coefficient		
With respect to	Θ	Weights of DeepGeo	151 585	
	P_z	Control points' z coordinates		720
		Twist function		7
	α	Angle of attack	1	1
Total design variables			151 586	728
Subject to	$C_L = 0.5$	Lift coefficient constraint	1	1
	$C_M \geq -0.17$	Moment coefficient constraint	1	1
	$t \geq 0.25\, t_{\text{original}}$	Minimum thickness constraints		750
	$Vol \geq Vol_{\text{original}}$	Minimum volume constraint	1	1
	$\Delta V^{\text{TE}} = 0$	Trailing edge constraint	1	
	$\Delta V_x^{\text{LE}} = 0,\ \Delta V_y^{\text{LE}} = 0$	Leading edge constraint	1	
	$\delta \mathbf{v}^{\text{LE,root}} = 0$	Fixed-wing root incidence constraint	1	
	$\Delta P_z^{\text{TE,upper}} = -\Delta P_z^{\text{TE,lower}}$	Fixed trailing-edge constraints		15
	$\Delta P_z^{\text{LE,upper,root}} = -\Delta P_z^{\text{LE,lower,root}}$	Fixed-wing root incidence constraint		1
	$2.0 \leq \alpha \leq 4.0$	Angle of attack constraint	1	1
Total constraints			7	770
Need value range limits for each DV?			NO	YES

their displacements are denoted as ΔP. Both the local shape function and the twist function are employed. The baseline optimization uses the MACH-Aero framework, implementing the parameterization with pyGeo [5], deforming the computational mesh with IDWarp [23], and optimizing with IPOPT in pyOptSparse [61].

D. Results and Analysis

Quantitative results are presented in Table 3. A direct comparison of minimized drags show that the DeepGeo parameterization is quantitatively comparable to the best-tuned FFD.

Figure 7 demonstrates the optimization history with DeepGeo. The optimization includes two steps. First, DeepGeo parameterizes the initial CRM wing by self-initialization, similar to the first case study on optimizing a circle. With the first 300 iterations, the optimization increases the wing's lift-over-drag ratio (L/D) from 22.5 to 23.8. Once the optimization converges, we reparameterize the partially optimized wing by self-initialization again with a new DeepGeo model and continue the optimization. The wing's L/D reaches a maximum of 25.0 and then slowly decreases. The optimal result is collected at the 320th iteration.

A comparison of the pressure distributions and shock wave shapes is shown in Figure 8. The shock waves in both optimized results have been mostly reduced. The parallel spacing of pressure contour lines in both optimized results can be clearly observed, indicating improved aerodynamic performance, though they exhibit differences. Figure 9 compares the optimized wings with slices at different positions along the wing span. FFD and DeepGeo lead to different optimized shapes: FFD's result shows very sharp leading edges and significantly redistributed thickness along the wing span. DeepGeo's result shows less variation from the initial shape. The leading edges are less sharp, the trailing edges better satisfy the geometric constraint, and the wing tip avoids becoming too thin.

The results show that, under strict geometric constraint settings, DeepGeo achieves comparable aerodynamic

Table 3 CRM wing optimization results of the Case Study II. One count for C_D equals to 10^{-4}.

Parameterization	Final C_D (counts)	Final C_L	Final C_M	Final α
DeepGeo	200.05	0.5000	-0.1717	2.08
FFD	199.03	0.5000	-0.1700	2.14

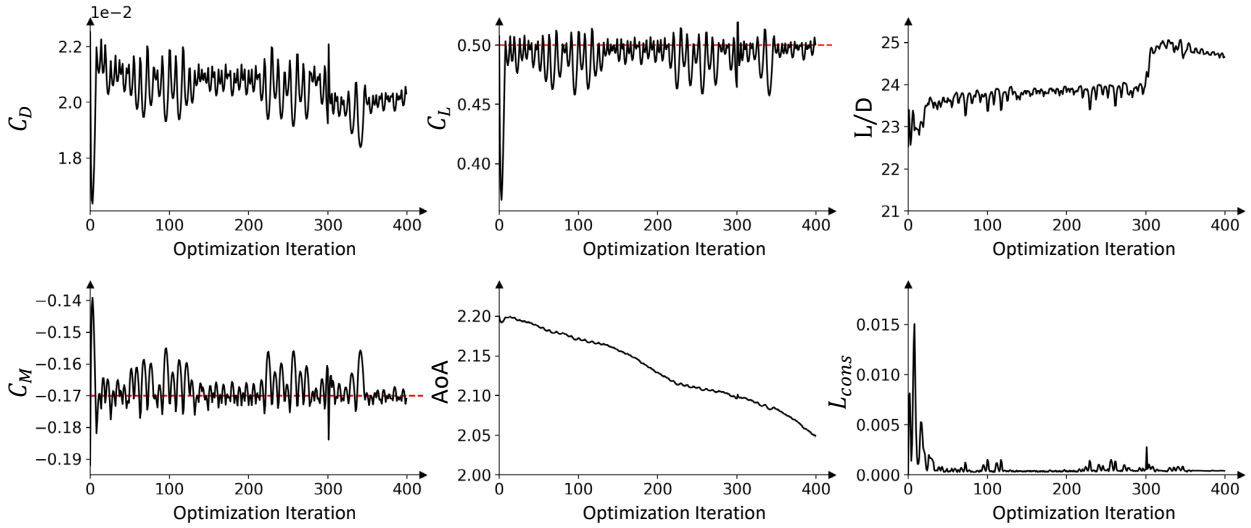

Fig. 7 The optimization history of Case Study II. The dashed line in C_L and C_M figures demonstrates the optimization objectives.

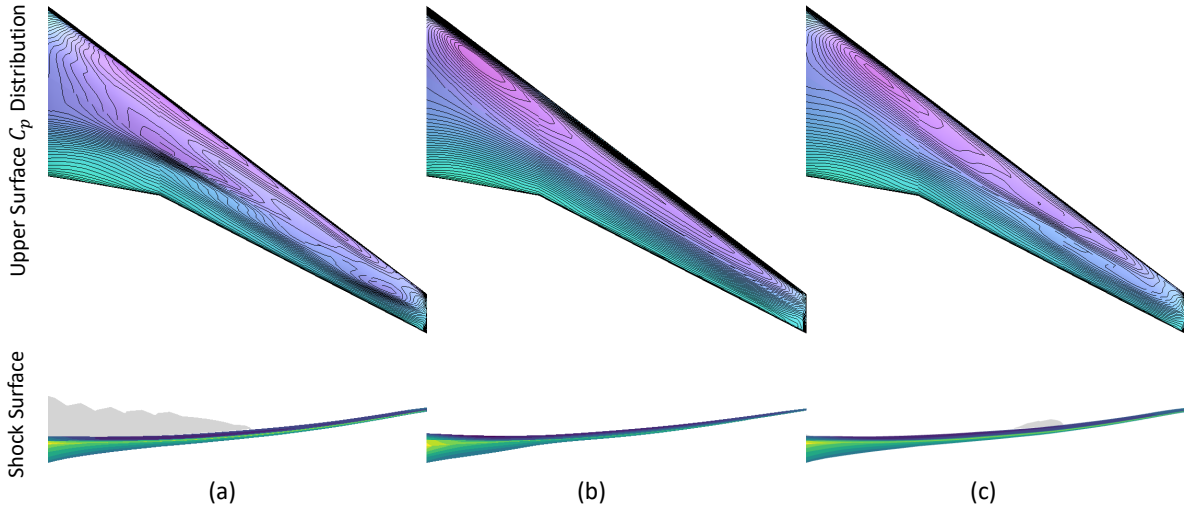

Fig. 8 A comparison of Case Study II on the pressure coefficient distributions and shock shapes of (a) the initial CRM wing, (b) the wing optimized with 192-point FFD, and (c) the wing optimized with DeepGeo.

performance with FFD. The DeepGeo-based optimal design is found in a narrow design space close to the baseline design, which may be due to the local convergence in solving the high-dimensional gradient-based optimization problem. Nevertheless, this design is preferable for industry applications because the off-design performance is well preserved with minor shape modifications from the initial design. In contrast, FFD-based optimization significantly reduces the leading edge to minimize on-design drag, which severely impacts low-speed aerodynamic performance [64], affecting aircraft landing and takeoff. Thus, optimization using DeepGeo is beneficial than using FFD when starting from a well-design conceptual baseline geometry, facilitates a more controllable ASO pipeline.

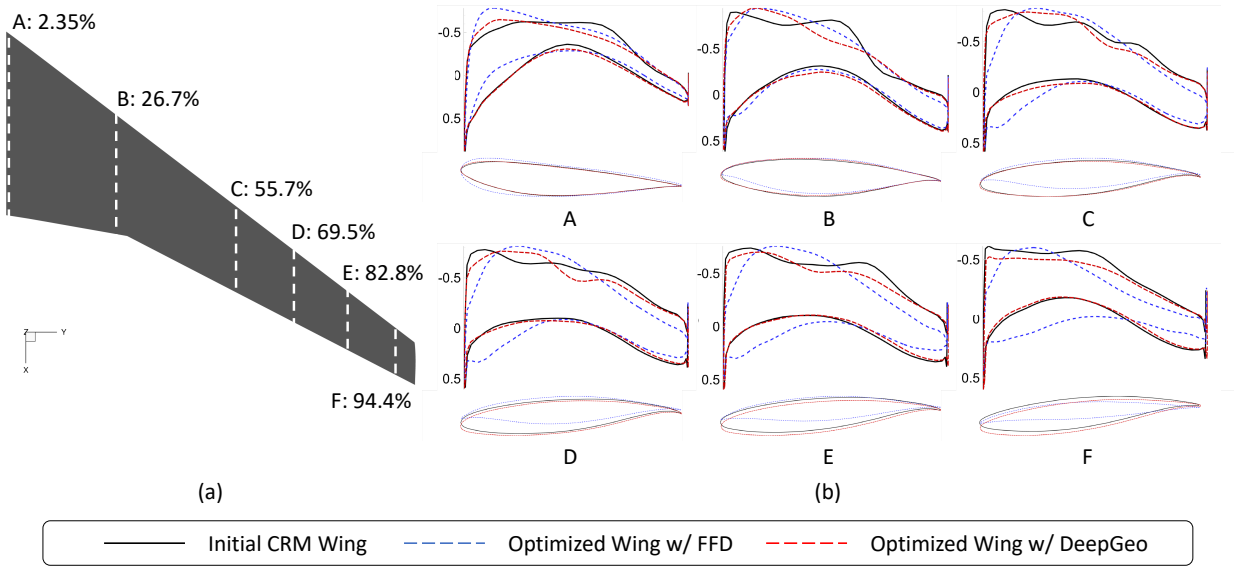

Fig. 9 A comparison of Case Study II on sliced geometries, including (a) the positions of slices, and (b) the shape variation and pressure coefficient distribution of each slice.

VII. Case Study III: the Blended-Wing-Body Aircraft Optimization

This case study demonstrates DeepGeo's capability in handling more complicated 3D geometry without tunning.

A. Problem Formulation

A Blended-Wing-Body (BWB) aircraft integrates the wing and fuselage into a single, seamless structure, aiming to improve aerodynamic efficiency, reduce drag, and enhance fuel economy. The initial BWB geometry, shown in Figure 10, is similar in planform to the first-generation Boeing BWB design with a capacity of 800 passengers [65]. This geometry has a span of 280 feet and a total length of 144 feet.

The objective of this case study is to minimize the drag coefficient C_D with the lift coefficient C_L constrained to 0.20056, and the pitching moment coefficient C_M constrained to 0. This single-point optimization problem is conducted under fully turbulent flow conditions at a Mach number of 0.85, Reynolds number of 5×10^6 and an initial angle of attack of 0.58°. The optimization is performed on both the L2, which has 1 142 505 cells. For geometric constraints,

Table 4 Aerodynamic shape optimization task specifications for the Case Study III.

Objectives	Functions/Variables	Description	DeepGeo Quantity	FFD Quantity
Minimize	C_D	Drag coefficient		
With respect to	Θ	Weights of DeepGeo	151 585	
	P_z	Control points' z coordinates		240
	α	Angle of attack	1	1
Total design variables			151 586	241
Subject to	$C_L = 0.20056$	Lift coefficient constraint	1	1
	$C_M = 0$	Moment coefficient constraint	1	1
	$t \geq 0.01\ t_{\text{original}}$	Minimum thickness constraints		750
	$Vol \geq Vol_{\text{original}}$	Minimum volume constraint	1	1
	$0 \leq \alpha \leq 2.5$	Angle of attack constraint	1	1
Total constraints			7	754
Need value range limits for each DV?			NO	YES

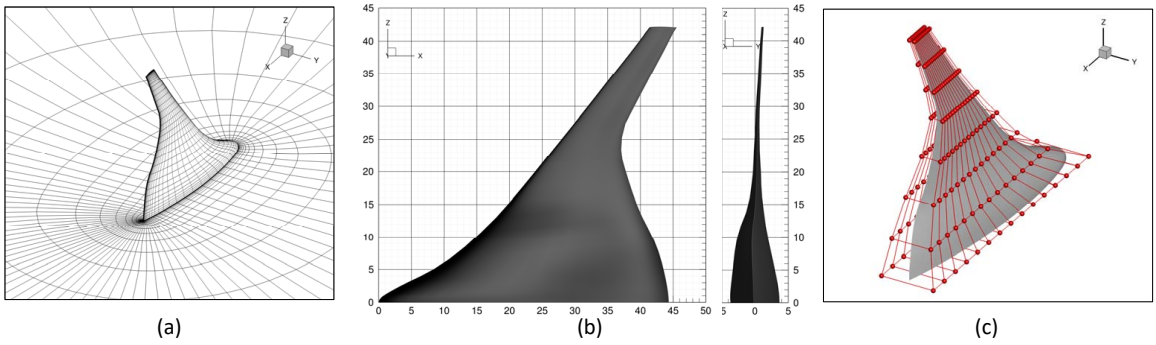

Fig. 10 The geometric setups for Case Study III, including: (a) the template mesh for DeepGeo, (b) the initial BWB aircraft geometry and (c) the 240-point FFD setting.

Table 5 BWB aircraft optimization results of the Case Study III.

Parameterization	Final C_D (counts)	Final C_L	Final C_M	Final α
DeepGeo	86.42	0.200 72	9.8807×10^{-4}	0.60
FFD	87.05	0.200 56	2.4675×10^{-9}	2.47

the aircraft's volume must be greater than or equal to its initial value. Shape changes are only allowed along the z axis to keep the aircraft's projected area constant. Table 4 summarizes the optimization problem.

B. Configurations of Deep Geometric Mapping Model

DeepGeo uses the L2 grid as template mesh. Similarly, DeepGeo parameterizes the BWB aircraft geometry through self-initialization, learning to generate all-zero deformation. DeepGeo's physical objective is defined as:

$$O_{CFD} = |C_D| + |C_L - 0.5| + |C_M| \ . \tag{16}$$

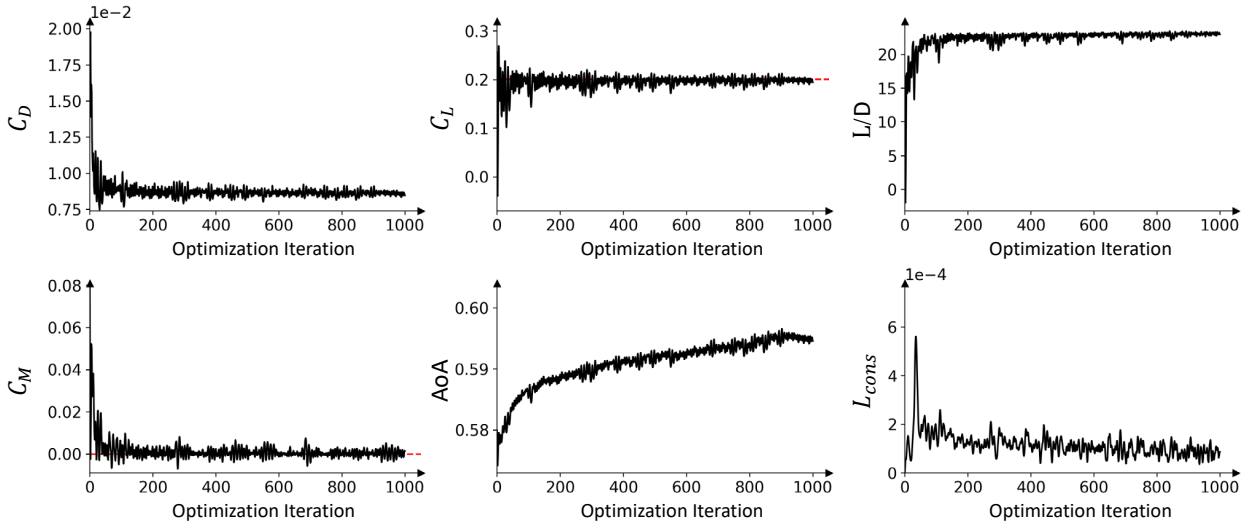

Fig. 11 The optimization history of Case Study III. The dashed line in C_L and C_M figures demonstrates the optimization objectives.

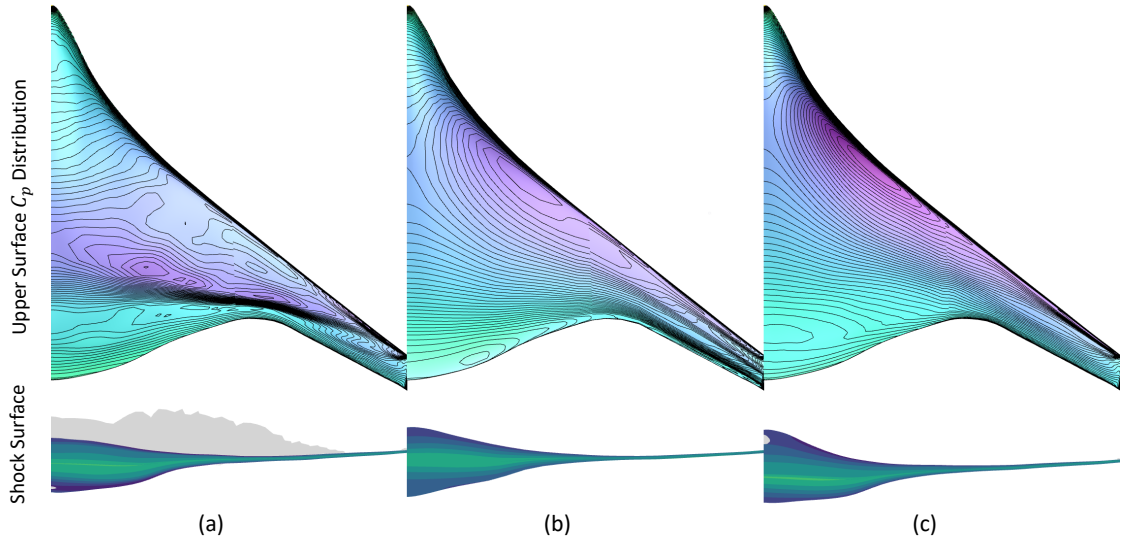

Fig. 12 A comparison of Case Study III on the pressure coefficient distributions and shock shapes of (a) the initial CRM wing, (b) the wing optimized with 240-point FFD, and (c) the wing optimized with DeepGeo.

In this case study, we enforce DeepGeo only generates shape variations along the z axis to follow the allowed changes defined by the optimization problem without the use of extra geometric constraints for the leading edge (LE) or trailing edge (TE). It is done by setting the x and y components of $F_\Theta(\hat{V})$ to 0. The only constraint relates to the total volume:

$$\mathcal{L}_{cons} = \max\left(Vol(V^S) - Vol_{\text{original}}(S), 0\right)^2 . \tag{17}$$

C. Configurations of Free-Form Deformation Baseline

The FFD configuration follows the well-established settings proposed by Lyu and Martins [66]. This configuration uses 240 control points, as shown in Figure 10(c). A minimal thickness constraint is added to avoid the crashes caused by the negative volume error during optimization. The FFD baseline case uses IPOPT implemented in pyOptSparse [61] for optimization.

D. Results and Analysis

The optimization history is shown in Figure 11. Quantitative comparisons of the optimization results are presented in Table 5. The DeepGeo-based optimized result demonstrates comparable aerodynamic performance to the state-of-the-art FFD-based one. Figure 12 compares the pressure distribution and the shock wave shapes. The optimal design solved using FFD even exhibits a double-shock on the wing tip, which should be avoided in practical design. In contrast, the design found by DeepGeo-based optimization has much weaker shock waves near the tip.

Figure 13 further analyze the geometry variations. The DeepGeo-based result shows more drastic changes in the span-wise bending and is less tilted upwards, while the shape variation on each slice is less significant. Similar to the result in Case Study II, the FFD-based result has very sharp leading edges.

The comparison concludes that, under less constrained settings, DeepGeo provides larger deformation freedom and thus a wider design space for exploration. In contrast, to enable FFD with a similar capability, users have to implement various global design functions by hand, such as sweep and dihedral, while restricting the value range of each design variable more cautiously to avoid failed optimization, which adds even more human engineering efforts.

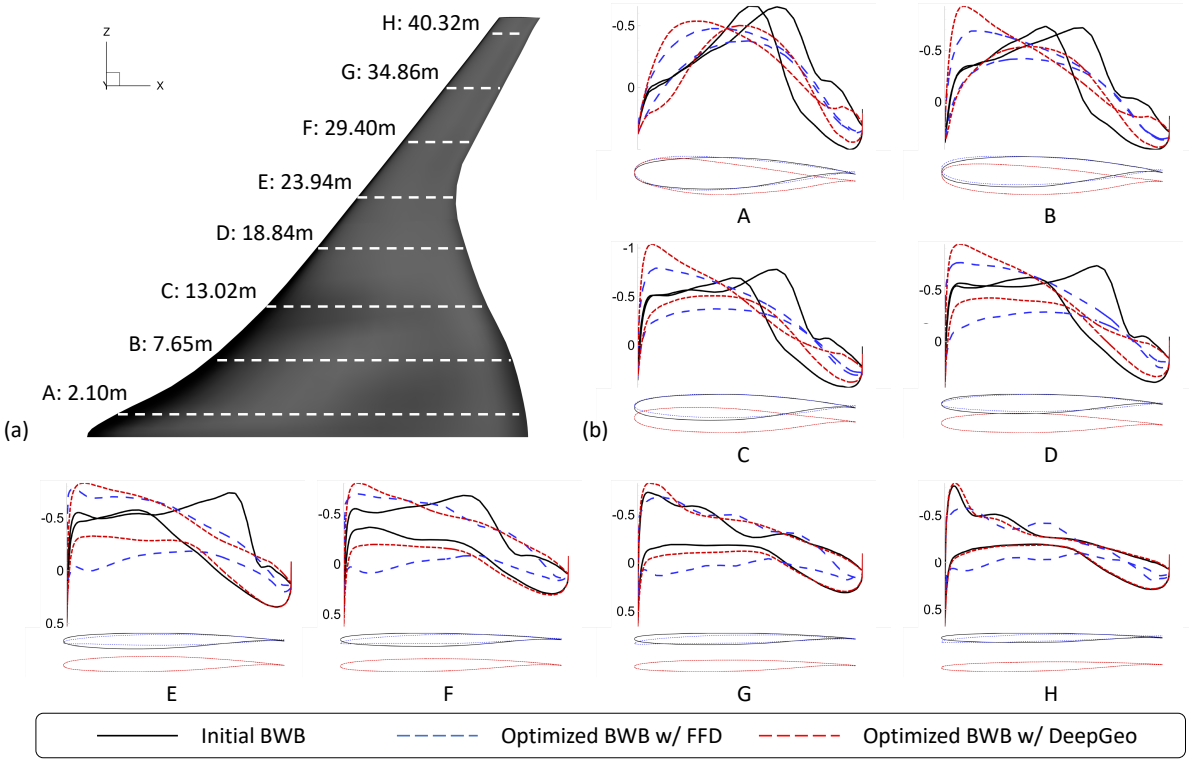

Fig. 13 A comparison of Case Study III on sliced geometries, including (a) the positions of slices, and (b) the shape variation and pressure coefficient distribution of each slice.

VIII. Conclusion

In this paper, we introduced the Deep Geometric Mapping (DeepGeo) model, a fully automated neural-network-based parameterization method for aerodynamic shape optimization (ASO). DeepGeo addresses key limitations of traditional parameterization methods by eliminating the need for extensive manual tuning and integrating volumetric mesh deformation, thus reducing implementation complexity and making the ASO pipeline more accessible. DeepGeo is based on deep geometric learning techniques but does not rely on large-scale training datasets.

Through multiple case studies, including 2D circle-to-airfoil optimization, 3D CRM wing optimization, and 3D Blended-Wing-Body aircraft optimization, we demonstrated the effectiveness and robustness of DeepGeo, showing its comparable performance to state-of-the-art free-form deformation methods and effectiveness in addressing their shortness. The results highlight DeepGeo's capability to provide significant shape deformation freedom, maintain global shape smoothness, and ensure the effectiveness of optimization.

In overall, DeepGeo offers a promising solution for complex geometric parameterization in the field of ASO. Future work will focus on further refining DeepGeo and exploring its applications in other ASO cases.

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. Rhea P. Liem and Aobo Yang are supported by the Hong Kong Research Grant Council General Research Fund (Project No. 16210621).

References

[1] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," Journal of Aircraft, Vol. 15, No. 7, 1978, pp. 407–412. https://doi.org/10.2514/3.58379.

[2] Kulfan, B. M., "Universal Parametric Geometry Representation Method," Journal of Aircraft, Vol. 45, No. 1, 2008, pp. 142–158. https://doi.org/10.2514/1.29958.

[3] Sederberg, T., and Parry, S., "Free-Form Deformation of Solid Geometric Models," ACM SIGGRAPH, Vol. 20, No. 4, 1986. https://doi.org/10.1145/15886.1590.

[4] Lamousin, H. J., and Jr., W. N. W., "NURBS-based free-form deformations," Computer Graphics and Applications, Vol. 14, No. 6, 1994, pp. 59–65. https://doi.org/10.1109/38.329096.

[5] Kenway, G., Kennedy, G., and Martins, J. R. R. A., "A CAD-Free Approach to High-Fidelity Aerostructural Optimization," 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, Texas, USA, September 13-15, 2010. https://doi.org/10.2514/6.2010-9231.

[6] Masters, D. A., Taylor, N. J., Rendall, T. C. S., Allen, C. B., and Poole, D. J., "Geometric Comparison of Aerofoil Shape Parameterization Methods," AIAA Journal, Vol. 55, No. 5, 2017, pp. 1575–1589. https://doi.org/doi.org/10.2514/1.J054943.

[7] Vuruskan, A., and Hosder, S., "Impact of Turbulence Models and Shape Parameterization on Robust Aerodynamic Shape Optimization," Journal of Aircraft, Vol. 56, No. 3, 2019, pp. 1099–1115. https://doi.org/10.2514/1.C035039.

[8] Song, W., and Keane, A., "A Study of Shape Parameterisation Methods for Airfoil Optimisation," 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004. https://doi.org/10.2514/6.2004-4482.

[9] Jameson, A., "Optimum aerodynamic design using CFD and control theory," Computational Fluid Dynamics Conference, 1995, p. 1729.

[10] Hansen, N., "The CMA Evolution Strategy: A Tutorial," arXiv Preprint, 2016.

[11] Wei, Z., Fua, P., and Bauerheim, M., "Automatic Parameterization for Aerodynamic Shape Optimization via Deep Geometric Learning," AIAA AVIATION Forum, 2023, p. 3471. https://doi.org/10.2514/6.2023-3471.

[12] Jameson, A., "Aerodynamic Design via Control Theory," Journal of Scientific Computing, Vol. 3, 1988, pp. 233–260. https://doi.org/10.1007/BF01061285.

[13] Braibant, V., and Fleury, C., "Shape optimal design using B-splines," Computer Methods in Applied Mechanics and Engineering, Vol. 44, No. 3, 1984, pp. 247–267. https://doi.org/10.1016/0045-7825(84)90132-4.

[14] Farin, G., Curves and Surfaces for Computer Aided Geometric Design, Academic Press Professional, Inc., USA, 1990. https://doi.org/10.1016/C2009-0-22351-8.

[15] Bloor, M. I. G., and Wilson, M. J., "Efficient parametrization of generic aircraft geometry," Journal of Aircraft, Vol. 32, No. 6, 1995, pp. 1269–1275. https://doi.org/10.2514/3.46874.

[16] Smith, R., Bloor, M., Wilson, M., and Thomas, A., "Rapid airplane parametric input design (RAPID)," 12th Computational Fluid Dynamics Conference, 1995. https://doi.org/10.2514/6.1995-1687.

[17] Pickett, R. M., Rubinstein, M., and Nelson, R., "Automated Structural Synthesis Using a Reduced Number of Design Coordinates," AIAA Journal, Vol. 11, No. 4, 1973, pp. 489–494. https://doi.org/10.2514/3.50489.

[18] Hager, J., Eyi, S., and Lee, K., "A multi-point optimization for transonic airfoil design," 4th Symposium on Multidisciplinary Analysis and Optimization, 1992. https://doi.org/10.2514/6.1992-4681.

[19] Sobieczky, H., "Parametric Airfoils and Wings," Recent Development of Aerodynamic Design Methodologies: inverse Design and Optimization, 1999, pp. 71–87.

[20] Barr, A. H., "Global and Local Deformations of Solid Primitives," ACM SIGGRAPH, Vol. 18, No. 3, 1984, pp. 21–30. https://doi.org/10.1145/964965.808573.

[21] Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark," AIAA Journal, Vol. 53, No. 4, 2015, pp. 968–985. https://doi.org/10.2514/1.J053318.

[22] Wu, N., Mader, C., and Martins, J. R., Sensitivity-based Geometric Parameterization for Aerodynamic Shape Optimization, 2022. https://doi.org/10.2514/6.2022-3931.

[23] Secco, N. R., Kenway, G. K. W., He, P., Mader, C., and Martins, J. R. R. A., "Efficient Mesh Generation and Deformation for Aerodynamic Shape Optimization," AIAA Journal, Vol. 59, No. 4, 2021, pp. 1151–1168. https://doi.org/10.2514/1.J059491.

[24] Ceze, M., Hayashi, M., and Volpe, E., "A Study of the CST Parameterization Characteristics," 27th AIAA Applied Aerodynamics Conference, 2009. https://doi.org/10.2514/6.2009-3767.

[25] Barron, A., "Universal approximation bounds for superpositions of a sigmoidal function," IEEE Transactions on Information Theory, Vol. 39, No. 3, 1993, pp. 930–945. https://doi.org/10.1109/18.256500.

[26] Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., and Liao, Q., "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review," International Journal of Automation and Computing, Vol. 14, 2017, pp. 503–519. https://doi.org/10.1007/s11633-017-1054-2.

[27] Li, J., Du, X., and Martins, J. R., "Machine learning in aerodynamic shape optimization," Progress in Aerospace Sciences, Vol. 134, 2022, p. 100849. https://doi.org/10.1016/j.paerosci.2022.100849.

[28] Robinson, G. M., and Keane, A. J., "Concise Orthogonal Representation of Supercritical Airfoils," Journal of Aircraft, Vol. 38, No. 3, 2001, pp. 580–583. https://doi.org/10.2514/2.2803.

[29] Poole, D. J., Allen, C. B., and Rendall, T. C. S., "Metric-Based Mathematical Derivation of Efficient Airfoil Design Variables," AIAA Journal, Vol. 53, No. 5, 2015, pp. 1349–1361. https://doi.org/10.2514/1.J053427.

[30] Li, J., Bouhlel, M. A., and Martins, J. R. R. A., "Data-Based Approach for Fast Airfoil Analysis and Optimization," AIAA Journal, Vol. 57, No. 2, 2019, pp. 581–596. https://doi.org/10.2514/1.J057129.

[31] Kedward, L., Allen, C. B., and Rendall, T., "Towards Generic Modal Design Variables for Aerodynamic Shape Optimisation," AIAA Scitech Forum, 2020. https://doi.org/10.2514/6.2020-0543.

[32] Constantine, P. G., Dow, E., and Wang, Q., "Active Subspace Methods in Theory and Practice: Applications to Kriging Surfaces," SIAM Journal on Scientific Computing, Vol. 36, No. 4, 2014, pp. A1500–A1524. https://doi.org/doi.org/10.1137/13091613.

[33] Li, J., Cai, J., and Qu, K., "Surrogate-Based Aerodynamic Shape Optimization with the Active Subspace Method," Structural and Multidisciplinary Optimization, Vol. 59, No. 2, 2019, pp. 403–419. https://doi.org/10.1007/s00158-018-2073-5.

[34] Lukaczyk, T. W., Constantine, P., Palacios, F., and Alonso, J. J., "Active Subspaces for Shape Optimization," Multidisciplinary Design Optimization Conference, 2014. https://doi.org/10.2514/6.2014-1171.

[35] Namura, N., Shimoyama, K., and Obayashi, S., "Kriging surrogate model with coordinate transformation based on likelihood and gradient," Journal of Global Optimization, Vol. 68, No. 3, 2017, pp. 827–849. https://doi.org/10.1007/s10898-017-0516-y.

[36] Grey, Z. J., and Constantine, P. G., "Active Subspaces of Airfoil Shape Parameterizations," AIAA Journal, Vol. 56, No. 5, 2018, pp. 2003–2017. https://doi.org/10.2514/1.J056054.

[37] Bauerheim, M., Ndiaye, A., Constantine, P., Moreau, S., and Nicoud, F., "Symmetry breaking of azimuthal thermoacoustic modes: the UQ perspective," Journal of Fluid Mechanics, Vol. 789, 2016, p. 534–566. https://doi.org/10.1017/jfm.2015.730.

[38] Payne, L. E., and Weinberger, H. F., "An optimal Poincaré inequality for convex domains," Archive for Rational Mechanics and Analysis, Vol. 5, 1960, pp. 286–292. https://doi.org/10.1007/BF00252910.

[39] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U., "When is "nearest neighbor" meaningful?" International Conference on Database Theory, 1999, pp. 217–235. https://doi.org/10.1007/3-540-49257-7_15.

[40] Viswanath, A., Forrester, A. I. J., and Keane, A. J., "Dimension Reduction for Aerodynamic Design Optimization," AIAA Journal, Vol. 49, No. 6, 2011, pp. 1256–1266. https://doi.org/10.2514/1.J050717.

[41] 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.

[42] Chen, W., Chiu, K., and Fuge, M. D., "Airfoil Design Parameterization and Optimization Using Bézier Generative Adversarial Networks," AIAA Journal, Vol. 58, No. 11, 2020, pp. 4723–4735. https://doi.org/10.2514/1.J059317.

[43] Li, J., Zhang, M., Martins, J. R. R. A., and Shu, C., "Efficient Aerodynamic Shape Optimization with Deep-Learning-Based Geometric Filtering," AIAA Journal, Vol. 58, No. 10, 2020, pp. 4243–4259. https://doi.org/10.2514/1.J059254.

[44] Li, J., and Zhang, M., "On Deep-Learning-Based Geometric Filtering in Aerodynamic Shape Optimization," Aerospace Science and Technology, Vol. 112, 2021, p. 106603.

[45] Sheffer, A., Praun, E., and Rose, K., "Mesh Parameterization Methods and Their Applications," Foundations and Trends® in Computer Graphics and Vision, Vol. 2, No. 2, 2006, pp. 105–171. https://doi.org/10.1561/0600000011.

[46] Floater, M. S., and Hormann, K., "Surface Parameterization: a Tutorial and Survey," Advances in Multiresolution for Geometric Modelling, 2005, pp. 157–186.

[47] Sumner, R. W., and Popović, J., "Deformation Transfer for Triangle Meshes," ACM Transactions on Graphics, Vol. 23, No. 3, 2004, pp. 399–405. https://doi.org/10.1145/1015706.1015736.

[48] Sorkine, O., and Alexa, M., "As-Rigid-As-Possible Surface Modeling," Geometry Processing, 2007. https://doi.org/10.2312/SGP/SGP07/109-116.

[49] Lévy, B., Petitjean, S., Ray, N., and Maillot, J., "Least Squares Conformal Maps for Automatic Texture Atlas Generation," ACM Transactions on Graphics, Vol. 21, No. 3, 2002, pp. 362–371. https://doi.org/10.1145/566654.566590.

[50] Schreiner, J., Asirvatham, A., Praun, E., and Hoppe, H., "Inter-Surface Mapping," ACM Transactions on Graphics, Vol. 23, No. 3, 2004, pp. 870–877. https://doi.org/10.1145/1015706.1015812.

[51] Smith, J., and Schaefer, S., "Bijective Parameterization with Free Boundaries," ACM Transactions on Graphics, Vol. 34, No. 4, 2015. https://doi.org/10.1145/2766947.

[52] Terzopoulos, D., Witkin, A., and Kass, M., "Symmetry-Seeking Models and 3D Object Reconstruction," International Journal of Computer Vision, Vol. 1, 1987, pp. 211–221.

[53] Ulyanov, D., Vedaldi, A., and Lempitsky, V., "Deep Image Prior," Conference on Computer Vision and Pattern Recognition, 2018, pp. 9446–9454.

[54] Williams, F., Schneider, T., Silva, C. T., Zorin, D., Bruna, J., and Panozzo, D., "Deep Geometric Prior for Surface Reconstruction," Conference on Computer Vision and Pattern Recognition, 2019, pp. 10130–10139. https://doi.org/10.1109/CVPR.2019.01037.

[55] Hanocka, R., Metzer, G., Giryes, R., and Cohen-Or, D., "Point2Mesh: A Self-Prior for Deformable Meshes," ACM SIGGRAPH, Vol. 39, No. 4, 2020. https://doi.org/10.1145/3386569.3392415.

[56] Knupp, P., "Remarks on Mesh Quality," Tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2007.

[57] Ba, J., Kiros, J., and Hinton, G., "Layer Normalization," arXiv Preprint, 2016.

[58] Fukushima, K., "Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements," IEEE Transactions on Systems Science and Cybernetics, Vol. 5, No. 4, 1969, pp. 322–333. https://doi.org/10.1109/TSSC.1969.300225.

[59] Mader, C. A., Kenway, G. K. W., Yildirim, A., and Martins, J. R. R. A., "ADflow—An open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization," Journal of Aerospace Information Systems, 2020. https://doi.org/10.2514/1.I010796.

[60] Morgan, N., and Bourlard, H., "Generalization and Parameter Estimation in Feedforward Netws: Some Experiments," Neural Information Processing Systems, 1989. URL https://api.semanticscholar.org/CorpusID:18821787.

[61] Wu, E., Kenway, G., Mader, C. A., Jasa, J., and Martins, J. R. R. A., "pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems," Journal of Open Source Software, Vol. 5, No. 54, 2020, p. 2564. https://doi.org/10.21105/joss.02564.

[62] Zhang, C., and Chen, T., "Efficient feature extraction for 2D/3D objects in mesh representation," International Conference on Image Processing, Vol. 3, 2001, pp. 935–938.

[63] Hwang, J. T., Jasa, J. P., and Martins, J. R. R. A., "High-Fidelity Design-Allocation Optimization of a Commercial Aircraft Maximizing Airline Profit," Journal of Aircraft, Vol. 56, No. 3, 2019, pp. 1164–1178. https://doi.org/doi.org/10.2514/1.C035082.

[64] Li, J., He, S., and Martins, J. R., "Data-driven constraint approach to ensure low-speed performance in transonic aerodynamic shape optimization," Aerospace Science and Technology, Vol. 92, 2019, p. 536–550. https://doi.org/10.1016/j.ast.2019.06.008.

[65] Liebeck, R. H., "Design of the Blended Wing Body Subsonic Transport," Journal of Aircraft, Vol. 41, No. 1, 2004, pp. 10–25. https://doi.org/10.2514/1.9084.

[66] Lyu, Z., and Martins, J. R. R. A., "Aerodynamic Design Optimization Studies of a Blended-Wing-Body Aircraft," Journal of Aircraft, Vol. 51, No. 5, 2014, pp. 1604–1617. https://doi.org/10.2514/1.C032491.