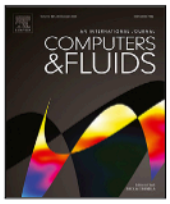
Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid



Highlights

A multi-fidelity workflow for conceptual design using Graph Neural Network-based field prediction surrogate model

Computers and Fluids xxx (xxxx) xxx

Yiren Shen*, Jacob T. Needels, Juan J. Alonso

- Designed a GNN-based surrogate model (`MF-VortexNet`) that leverages CFD to augment VLM predicted pressure field.
- Integrated the surrogate model into a multi-fidelity conceptual design workflow.
- Proposed surrogate model generalizes well across unseen geometries in the design space.
- Achieved a 5× reduction in lift and moment NRMSE relative to VLM, using CFD results as reference.
- Investigated the surrogate model's feature capturing mechanism through a latent space ablation study.

Graphical abstract and Research highlights will be displayed in online search result lists, the online contents list and the online article, but will not appear in the article PDF file or print unless it is mentioned in the journal specific style requirement. They are displayed in the proof pdf for review purpose only.

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

A multi-fidelity workflow for conceptual design using Graph Neural Network-based field prediction surrogate model☆

Yiren Shen *, Jacob T. Needels, Juan J. Alonso

Department of Aeronautics and Astronautics, Stanford University, 496 Lomita Mall, Stanford, 94305, CA, USA

ARTICLE INFO

Keywords:
Surrogate modeling
Aerodynamic design
Multi-fidelity analysis
Vortex lattice method
Computational fluid dynamics
Deep learning

ABSTRACT

Aircraft design relies on accurate aerodynamic predictions. High-fidelity (HF) methods, such as computational fluid dynamics (CFD), provide accurate predictions of aerodynamic loads but are computationally expensive for early-stage design. Conversely, low-fidelity (LF) methods, such as the vortex lattice method (VLM), offer rapid solutions with low computational cost but struggle to capture complex flow phenomena, thereby limiting their predictive accuracy. The conceptual design process thus presents a fidelity-cost trade-off, requiring a balance between HF and LF data in early design phases. This study proposes a multi-fidelity conceptual design analysis workflow that integrates a Graph Neural Network (GNN)-based physics-informed lightweight trainable surrogate model, Multi-Fidelity VortexNet (`MF-VortexNet`), into the VLM solver to augment the analysis fidelity of LF tools. The surrogate model learns corrections to LF panel-wise pressure field data using data-driven insights from HF CFD simulations, thus allowing accurate and efficient aerodynamic analyses on arbitrary quantities of interest. When the proposed workflow is used for performance assessment of various Delta wing geometries under low speed and high angle of attack, the surrogate model demonstrates strong prediction accuracy and generalizability across geometries and free-stream conditions, and it effectively captures the nonlinear flow features under geometric variations. Compared to using VLM alone, the multi-fidelity analysis workflow achieves a fivefold reduction in the normalized root mean square error for the predicted aerodynamic coefficients. A prediction mechanism explanation, leveraging the latent space ablation technique, is conducted to rationalize the model's predictive capabilities and provide guidance for future improvements in `VortexNet`-like surrogate modeling. These results indicate that the proposed multi-fidelity workflow has the potential to serve as a valuable tool for design fidelity improvement in conceptual design.

1. Introduction

Aircraft design involves computational modeling to predict and optimize vehicle aerodynamic performance. Accurate simulation can be accomplished using high-fidelity (HF) computational fluid dynamics (CFD) codes to solve the Navier–Stokes equations, a set of coupled nonlinear partial differential equations (PDEs), which govern viscous, compressible flow. Despite significant progress in CFD and high-performance computing hardware, CFD remains expensive not only due to its substantial computational cost, but also because of the intensive workflow associated with geometry preparation, mesh generation, and post-processing [1,2]. Such costs accumulate significantly in many-query applications, such as design optimization, where a large number of flow-field evaluations across a wide range of flight conditions and vehicle configurations are required. Consequently, exclusive reliance on CFD during the conceptual aircraft design phase becomes prohibitive. During this phase, geometry assets required for CFD analyses are often not available. Engineers and researchers frequently utilize low-fidelity (LF) methods, subject matter expert knowledge, and historical data to explore the design space and draft an initial design [3,4]. As the aircraft configuration matures, candidate designs are assessed and revised with higher fidelity computational and experimental methods, including CFD and wind tunnel testing. This process carries the risk that inaccuracies introduced by lower fidelity tools will result in a non-optimal configuration, requiring significant revision in later stages and adversely impacting development cost and schedule. Improved accuracy in conceptual-level design tools has the potential to improve prediction quality and confidence in configuration performance and also motivates the introduction of HF design tools earlier in the design cycle. However, the computational expense of applying HF methods across the joint design–mission space remains prohibitive. Consequently, strategies are

☆ This article is part of a Special issue entitled: 'fusing data and physics' published in Computers and Fluids.

* Corresponding author.
E-mail address: yrshen@stanford.edu (Y. Shen).

https://doi.org/10.1016/j.compfluid.2026.107153
Received 1 April 2025; Received in revised form 24 February 2026; Accepted 22 May 2026

needed to augment LF predictions with HF-informed accuracy while maintaining evaluation efficiency.

1.1. Motivation

Recent interest in multidisciplinary design optimization (MDO) [5,6] calls for a more closely integrated design-assessment cycle for aircraft development, and many conceptual design code suites have been developed to address this need, such as SUAVE [7,8]. These software tools provide aircraft design analysis and optimization frameworks that incorporate hierarchical multi-fidelity (MF) analyses across multiple disciplines to explore the design space and evaluate the resultant vehicle performance, including but not limited to flight dynamics, efficiency, structural dynamics, and handling qualities. Although HF data sources, including experimental results, can be included in this framework, conceptual design primarily relies on engineering-level models that use low- to high-fidelity simulations to approximate the true physical system for performance assessment. It must be understood that all models, including high-fidelity ones, include certain modeling assumptions and are therefore only approximations of the true physical phenomena. As such, epistemic errors arising from modeling assumptions propagate into the design cycle and may bias the conceptual design. To enable rapid design iteration, lower-fidelity tools with lower assessment costs are preferred during the conceptual design phase. At the lowest fidelity, pressure distribution and force and moment coefficients for subsonic and transonic aircraft are typically computed using a vortex lattice method (VLM) code [9–12] due to its relative simplicity in analysis setup and reduced computational cost [13]. The VLM models the pressure distribution on the aircraft based on potential flow theory, neglecting compressibility, viscous effects, and certain geometry aspects such as wing thickness.

Despite the relative popularity of the VLM, its underlying assumptions make it incapable of modeling certain physical phenomena and conceptual design use cases. For example, the conceptual design of Delta wing platforms at low-speed, incentivized by renewed interest in supersonic transport (SST) development [14], requires the accurate prediction of aerodynamic coefficients and pressure distribution to impose low-speed control and stability constraints [15,16]. During landing, SST vehicles encounter a free-stream at low speed and high angle of attack (AOA), resulting in complex, nonlinear flow physics around the Delta wing planform, such as vortex lift, vortex-boundary layer interactions, and flow separation [17]. Unfortunately, VLM prediction of aerodynamic coefficients and pressure distribution under such flow conditions is largely inaccurate because these nonlinear effects are not captured by the potential flow model. An option to improve VLM prediction accuracy is to incorporate theoretical or semi-empirical models, such as the Polhamus correction [18], into VLM solvers. However, this approach has drawbacks regarding accuracy and platform applicability: correction parameters need to be tuned for specific wing configurations and flow mechanisms to produce accurate predictions [19,20]. When a large design space is explored in a conceptual design task, validating the correction parameters themselves across the design space can be a cumbersome task.

Thus, to accurately evaluate the low-speed characteristics of Delta wing-based configurations using computational modeling, higher fidelity methods such as CFD need to be used. Methods such as detached-eddy simulation (DES) have demonstrated the ability to accurately model the vortical flow field and predict the location of vortex breakdown [21], thereby leading to accurate pressure distribution and stability estimation. Reynolds-averaged Navier–Stokes (RANS)-based CFD methods are also widely used [22–24]. The fidelity and computational cost of RANS lie between DES and VLM. Seraj and Martins demonstrated that RANS achieves satisfactory accuracy in the prediction of aerodynamic coefficients, including the pitching moment coefficient (C_M), for Delta wings at moderate AOAs during takeoff and landing [15]. However, embedding a CFD solver within the aerodynamic shape optimization (ASO) loop is computationally prohibitive, particularly during the conceptual design phase when high-fidelity geometries are not yet available and the configuration varies substantially across the design space.

The aforementioned fidelity-cost trade-off exists ubiquitously in every ASO problem. To reduce the evaluation cost in an optimization loop, rather than adding additional correction terms to LF models, surrogate modeling techniques are often used, such that the surrogate model approximates the behavior of HF CFD using a simplified mathematical model. When MF computational data are avilable, higher fidelity data can be used to calibrate lower fidelity methods' predictions, thereby achieving improved analysis fidelity while preserving the low evaluation cost of LF tools. Such frameworks are attractive because they are extensible to different problems with accuracy comparable to the highest fidelity analysis tool, given sufficient samples and training, thereby allowing faster evaluations, and they can be used as a drop-in enhancement of the current MDO workflow.

1.2. Background

One common surrogate modeling approach is to learn corrections to scalar-valued aerodynamic force and moment coefficients through Bayesian frameworks, such as through MF Gaussian Processes Regression (GPR), to synthesize multiple data sources [25]. Mukhopadhaya et al. demonstrated such an approach by generating multi-fidelity probabilistic aerodynamic databases for commercial aircraft simulation, using data from VLM evaluations, RANS CFD simulations, and wind tunnel experiments [26]. SUAVE has a built-in module to accommodate MF data sources and build surrogate models for design optimization purposes, demonstrating a good balance between computational efficiency and accuracy [8]. However, standard GPR is limited to scalar-valued quantities of interest (QOIs), making it challenging to apply to high-dimensional data like surface or volume fields. In many applications, accurate prediction of field data is necessary to resolve location-dependent phenomena, such as turbulent transition and flow separation, to enable the evaluation of arbitrary QOIs that are not prescribed a priori, and to support coupling among solvers across disciplines. The ability of the surrogate to produce field data predictions also aids in the interpretability of results, improving understanding of and confidence in the predictions.

Popular techniques for surrogate modeling of aerodynamic field data frequently utilize proper orthogonal decomposition (POD) [27]. By decomposing the flow-field into a set of spatial modes, POD permits a parameterized model of field variables based on the linear combination of a truncated set of basis vectors. Using regression to predict modal amplitudes as a function of input parameters enables a method for non-intrusive field surrogate modeling [28–30]. POD-kriging is a popular technique where POD coefficient amplitudes are predicted using kriging for modal regression, and has been previously applied to design applications, including the aerodynamic analysis of automobile configurations [28]. In that work, Mrosek et al. demonstrate the application of POD-kriging to the analysis of varying geometries utilizing a single fidelity of aerodynamic data. Recent approaches have also sought to integrate POD surrogate modeling approaches into multi-fidelity frameworks, including through the use of manifold alignment to combine data from different sources [31].

The linear projection approach of POD can have limited accuracy when modeling complex and nonlinear features with a limited number of modes, motivating exploration of nonlinear dimensionality reduction methods [32]. Such techniques, coupled with regression, have been explored for both single and multi-fidelity data sources [33,34]. Significant flexibility can be obtained for field representation by utilizing DNN architectures, which can combine the dimensionality reduction and regression functions into a single model with the ability to represent arbitrarily complex and nonlinear field quantities. In Shen et al., the ability to effectively learn the discrepancy between LF Euler and HF

RANS flow fields using a DNN architecture for a fixed geometry is demonstrated [2]. However, their use in design applications remains challenging because the surrogate must generalize across changing geometries. In this work, we examine the extension of multi-fidelity DNN architectures for design applications.

To adequately extend VLM predictions for the case of interest, analysis and optimization of Delta wing configurations at high AOAs, both a field prediction capability and strong geometric generalizability are required. To overcome limitations of existing methods, we explore approaches involving a lattice-wise vortex induced pressure modification based on CFD results. Instead of relying on semi-empirical models, we propose using a graph neural network (GNN) to learn a mapping that calibrates LF predictions using HF CFD results. Recent studies show that GNNs have strong potential for this type of mapping, offering superior performance compared to traditional methods [35–41]. Due to their graph nature, GNNs are highly effective at integrating and learning mappings between different fidelity data sources, including field data sampled at different spatial locations [36,42,43]. Additionally, complex geometries and physical constraints can be efficiently represented by a graph, even with underlying geometric variations, and thus GNNs demonstrate superior generalizability for geometry changes, data scalability, and efficiency [36,37]. These characteristics will be critical for MDO applications as the vehicle configuration, mesh discretization, free-stream conditions, and the QOIs may change during the design-assessment cycle.

1.3. Overview

Motivated by these research advancements, this study aims to enhance the accuracy of the VLM in conceptual aircraft design by proposing a GNN-based mapping function to correct the induced pressure difference predicted by the LF solver using HF CFD data. To showcase the workflow's significance in improving design fidelity for cases that VLM under-performs, as discussed in Section 1.1, the proposed workflow is tested with a benchmark case of a Delta wing at various AOAs. While parts of the results have been presented in previous conference proceedings [44,45], this paper revisits, consolidates, and expands upon those findings, providing a cohesive and comprehensive discussion of the workflow.

The paper is organized as follows. In Section 2, an overview of the MF analysis tools to generate the training dataset is given, along with the GNN architecture and training procedure. Section 3 discusses the dataset used to train and validate the surrogate model. Section 4 presents network prediction quality for aerodynamic loading across a range of vehicle geometries. A study of how input graph features are propagated through latent layers is conducted to better understand the relationship between network architecture and prediction quality. The paper concludes with an overview of results and directions for future work.

2. Methodology

To construct the proposed GNN-based MF surrogate model, pressure distribution data are used from two models of different fidelity. LF data are generated using the panel-wise pressure distribution computed by a VLM solver in the SUAVE [7,8] software package. For the HF data, we use the surface pressure coefficients computed by the SU2 CFD solver [46]. A brief summary of the physical and numerical models for each method is presented in Section 2.1 for VLM and Section 2.2 for CFD. The proposed multi-fidelity analysis framework integrated into SUAVE is presented in Section 2.3. Finally, the GNN surrogate model's network architecture and the associated training procedure are described in Section 2.4.

2.1. Low-fidelity solver: Vortex lattice method

The VLM solver used in SUAVE is based on VORLAX [11]. It models compressibility-corrected, inviscid, attached flows for both subsonic and supersonic conditions by using a panel method that represents the base geometry with infinitesimally thin, cambered or uncambered panels [47]. Each panel consists of a horseshoe vortex with trailing vortices extending to infinity and a bound vortex at the quarter-chord location. The control point $[x_c, y_c, z_c]$ is located at the three-quarter-chord point of each panel. The VORLAX implementation of VLM differs from typical textbook VLM [48] as multiple application-specific correction terms are introduced. To assist the discussion in later sections, we include a short summary of VLM. One should refer to Miranda's original manuscript for the exact algorithm used in VORLAX [11].

According to the Biot–Savart Law, the induced velocity from a vortex filament can be described as:

$$\mathbf{v} = \frac{\Gamma_n}{4\pi}\int_c \frac{d\mathbf{l}\times\mathbf{r}}{r^3}, \tag{1}$$

where Γ_n is the vortex filament strength, $d\mathbf{l}$ is an infinitesimal segment of the vortex filament, $\mathbf{r}$ is a vector pointing from the vortex filament to a point in space, r is the length of that vector, and c is the length of the vortex filament. For a horseshoe vortex with known endpoints A and B in space, the induced velocity vector $\mathbf{v}$ at any point C in space can be expressed as:

$$\mathbf{v}_{AB} = \frac{\Gamma_n}{4\pi}\left(\frac{\mathbf{r}_0}{|\mathbf{r}_1\times\mathbf{r}_2|}\left(\frac{\mathbf{r}_0\mathbf{r}_1}{|\mathbf{r}_0\mathbf{r}_1|} - \frac{\mathbf{r}_0\mathbf{r}_2}{|\mathbf{r}_0\mathbf{r}_2|}\right)\right), \tag{2}$$

where $\mathbf{r}_0$ is the vector from point A to B, $\mathbf{r}_1$ is the vector from point A to C, and $\mathbf{r}_2$ is the vector from point B to C. In Eq. (2), all vectors can be computed if the control point and vortex filament locations are known, thereby establishing a function between Γ_n and $\mathbf{v}$. In a system with multiple vortex filaments, the resultant induced velocity at any point in space is the linear superposition of all vortex filament contributions. For a system with N horseshoe vortices (panels), at the control point m, the induced velocity is then

$$\mathbf{v}_m = \sum_{n=1}^{N} C_{m,n}\Gamma_n, \tag{3}$$

where $C_{m,n}$ is the influence matrix computed from Eq. (2).

Applying the tangency boundary condition $\mathbf{n}\cdot\mathbf{v} = 0$, where $\mathbf{n}$ denotes the local unit normal vector, on both the body surface and the wake uniquely defines the system. These boundary conditions, together with the spatial correlations, can be conveniently summarized into a right-hand side vector (RHS), resulting in the final linear system for $\mathbf{\Gamma}$, the vector all vortex strengths at all filaments, as follows:

$$AIC\,\mathbf{\Gamma} = RHS, \tag{4}$$

where AIC is the aerodynamic influence coefficient matrix representing the induced velocity contributions from each vortex filament to each panel's control points, and RHS represents the free-stream flow contributions to the boundary conditions. The linear system can thus be solved for individual lattice vortex strengths.

Once the vortex filament strengths are solved, the pressure coefficient distribution is computed by

$$C_p = -\frac{2}{q_\infty^2}(U_\infty u + V_\infty v), \tag{5}$$

where U_∞ and V_∞ are the x and y components of the free-stream velocity vector of magnitude q_∞, and u and v are the induced velocity components. C_p is the field pressure coefficient ($C_p = (P - P_\infty)/q_\infty$). In our study, an infinitesimally thin panel is used such that the pressure difference between the lower and upper wing surfaces, ΔC_p, is the C_p computed from the previous equation.

Finally, ΔC_p is converted to forces and moments as follows: For each panel, the local loading coefficient is computed by multiplying ΔC_p by the panel area factor. The loading is then decomposed into normal and

axial force components, where the axial component is further corrected using a leading-edge thrust correction [49]. If the Polhamus suction correction is active [18], this leading-edge thrust is rotated by 90° to be normal to the leading-edge panel chord. The forces are then rotated from the local chord coordinate system to the body axes using the chord incidence and dihedral angles. The dimensional forces and moments for the vehicle are obtained by integrating the panel strip areas from the leading edge to the trailing edge and applying angle-of-attack and sideslip angle transformations.

2.2. High-fidelity solver: Reynolds-averaged Navier–Stokes

SU2 is a software suite for solving PDEs and PDE-constrained optimization problems on unstructured numerical grids [46,50,51]. To simulate compressible flows, SU2 solves the Navier–Stokes equations expressed in differential form as

$$\mathcal{R}(U) = \frac{\partial U}{\partial t} + \nabla \cdot \bar{F}^c(U) - \nabla \cdot \bar{F}^v(U, \nabla U) = 0. \tag{6}$$

The vector of conserved variables, U, can be expressed as

$$U = [\rho, \rho\bar{v}, \rho E]^{\mathsf{T}}, \tag{7}$$

where ρ is the fluid density, $\bar{v} = [u, v, w]^{\mathsf{T}}$ is the flow speed in a Cartesian system of reference, and E is the total energy per unit mass. The convective and viscous fluxes are given by

$$\bar{F}^c = \begin{bmatrix} \rho\bar{v} \\ \rho\bar{v} \otimes \bar{v} + \bar{\bar{I}}p \\ \rho E\bar{v} + p\bar{v} \end{bmatrix} \tag{8}$$

and

$$\bar{F}^v = \begin{bmatrix} \cdot \\ \bar{\bar{\tau}} \\ \bar{\bar{\tau}} \cdot \bar{v} + \kappa \nabla T \end{bmatrix} \tag{9}$$

respectively, where p is the static pressure, $\bar{\bar{\tau}}$ is the viscous stress tensor, T is the temperature, κ is the thermal conductivity, and μ is the viscosity. The viscous stress tensor can be expressed in vector notation as

$$\bar{\bar{\tau}} = \mu\left(\nabla\bar{v} + \nabla\bar{v}^T\right) - \mu\frac{2}{3}\bar{\bar{I}}\left(\nabla \cdot \bar{v}\right). \tag{10}$$

For a perfect gas with gas constant R and constant specific heat ratio γ, the ideal gas equation of state can be used to determine the pressure and temperature as $p = (\gamma - 1)\rho[E - 0.5(\bar{v} \cdot \bar{v})]$ and $T = p/(\rho R)$, respectively. Laminar viscosity, μ, is calculated using Sutherland's law and thermal conductivity can be computed as $\kappa = \mu c_p/Pr$, where c_p is the specific heat capacity at constant pressure and Pr is the Prandtl number, fixed at 0.72. Steady, turbulent flows can be simulated in SU2 by solving the RANS equations. In accordance with the Boussinesq hypothesis, the effective viscosity can be computed as the sum of the laminar viscosity (μ_d) and the turbulent viscosity (μ_t) as

$$\mu = \mu_d + \mu_t, \tag{11}$$

where the prediction of μ_t is dependent on the particular choice of turbulence model used.

In this work, the one-equation Spalart–Allmaras turbulence model with rotation correction is used to solve for the value of the turbulent viscosity [23,52]. The turbulence model is selected to achieve a balance between computational cost and adequate vortex effects quantification accuracy, as discussed in a previous study by Seraj and Martins [53]. All simulations employ the Jameson–Schmidt–Turkel scheme for convective discretization [54] and implicit time integration with an adaptive Courant–Friedrichs–Lewy number to converge the steady-state solution. The vehicle boundary condition is specified by a no-slip, adiabatic wall, and a characteristic-based free-stream is used for the far-field boundary. A Cauchy convergence criterion is used, such that the solver stops when the change in C_D for the wing over the previous 100 iterations becomes less than 1×10^{-4}.

2.3. Multi-fidelity workflow

The aforementioned HF and LF analyses are integrated in a multi-fidelity workflow, as illustrated in Fig. 1. In this chart, intermediate stages of data are presented in rounded boxes. Two types of data exist: data obtained in the online phase for MDO, that is, data obtained within the SUAVE environment, are colored in light gold, and data obtained in the offline phase, that is, the data generated for surrogate model training, are colored in light blue. Arrows connecting blocks represent the workflow steps, and annotations beside the arrows indicate the relevant tools that are used for each step.

The workflow starts with defining vehicle geometry and analysis settings. The vehicle is defined through existing SUAVE geometry classes, and is then analyzed using the `Vortex_Lattice` aerodynamics analysis module in SUAVE. Analysis parameters, such as lattice panel shape parameters including the number of panels in the chord-wise and span-wise directions and the Polhamus suction analogy flag, are also defined. In the current study, we use 30 panels in both the chord-wise and span-wise directions. The span-wise panel spacing uses a cosine spacing, while the chord-wise panel spacing uses even spacing. The Polhamus suction analogy is active for leading-edge panels.

Once the vehicle geometry and analyses setup is completed, the workflow diverges into two branches. In the online phase, a VLM analysis is carried out to obtain ΔC_p^{LF} and lattice panel geometry using algorithms defined in Section 2.1. Beyond standard SUAVE VLM output fields, the AIC matrix and RHS vectors in Eq. (4) are also recorded for surrogate model training [44]. In the offline phase, the SUAVE vehicle is exported to an `OpenVSP` [55] vehicle using SUAVE's `OpenVSP` application programming interface (API) for geometry export. Creating an `OpenVSP` geometry allows a user to create surface and volume meshes in a meshing software of one's choice. In the current study, we selected `Pointwise` to create CFD surface and volume meshes. `Pointwise` is selected due to the authors' familiarity with the software and its capability for scripting and generating high-quality RANS meshes using its `T-Rex` anisotropic boundary layer mesh capability. For all geometries in this study, a `Pointwise` script is used to control the mesh properties, including average nodal spacing, growth rate, and `T-Rex` growth parameters, to maintain a relatively consistent mesh across geometries.

Following its creation, the `Pointwise` volume mesh is exported in SU2 format for CFD computation. Details of the SU2 CFD configuration can be found in Section 2.2. Once the CFD computation converges, a surface projection as described in Section 3.2 is used to project the surface pressure C_p onto the lattice panels to obtain the HF pressure difference ΔC_p^{HF}. This ΔC_p^{HF} serves as the reference HF data for training the surrogate model. The offline phase concludes with surrogate model training, where a dataset generated from various geometries and flow conditions, as described in Section 2.4, is provided as input. Once training is complete, the surrogate model is ready to be used in online assessment.

In online assessment, for example, when one needs to evaluate the aerodynamics of new geometries or under new flow conditions, the vehicle definitions and analyses setup follows the same process as previously discussed in this section. However, in the `Vortex_Lattice` setting, one can choose whether a surrogate model should be used to augment the VLM ΔC_p^{LF}, as marked by a diamond-shaped decision point in the block diagram in Fig. 1. If a surrogate model is used, it takes in ΔC_p^{LF} and generates $\Delta C_p^{\mathrm{pred}}$ to replace the original ΔC_p^{LF}. Such replacement does not affect the existing functionalities in SUAVE's `Vortex_Lattice`, and thus all aerodynamic coefficients that are already defined in SUAVE can be evaluated. This also means that the existing design optimization methods defined in SUAVE can be used.

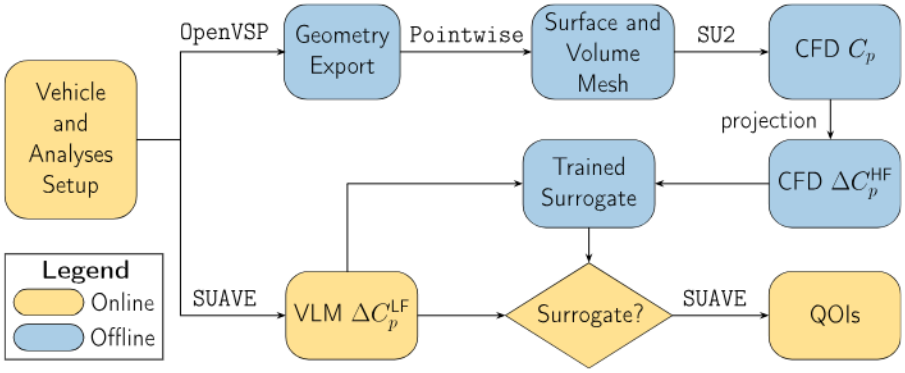


Fig. 1. Block diagram of the augmented aerodynamic coefficients estimation workflow integrated into the SUAVE environment. Each block represents an intermediate stage in the design workflow, with "online" stages colored gold and "offline" stages colored blue.

2.4. Surrogate modeling

A MF surrogate model is used to establish a mapping function, $\mathcal{F}$, between the LF field variables q^{LF} into their HF counter part q^{HF}. In general, such a mapping can be expressed as

$$q^{HF} \sim \mathcal{F}\left(q^{LF}, \omega_{\mathcal{L}}\right), \tag{12}$$

where $\omega_{\mathcal{L}}$ represents the weights and biases of the network under a specific loss function $\mathcal{L}$. In general, q may denote an arbitrary physical field, including surface quantities (e.g., pressure or skin-friction) or volumetric variables (e.g., velocity). In this study, we focus specifically on the pressure field over a reference plane. Such a mapping can be defined if the LF field variables q^{LF} exhibit an injective correspondence with the HF field variables q^{HF}, a hypothesis that has not been tested in this study. To construct such a surrogate model, a GNN-based U-net-like ML autoencoder is used. The network architecture and the training pipeline are presented in detail in the following sections.

2.4.1. Machine learning architecture

A graph $G = (V, E)$ consists of a node set $V = \{1, \ldots, N\}$ with cardinality N and an edge set E connecting pairs of nodes $i, j \in V$. A GNN is thus a model that performs machine learning on graphs to incorporate information about the structure of the graph to understand the relationship among nodes and edges [56]. This capability is classified into three prediction tasks: node, edge, and graph predictions. The node prediction capability is of particular interest to the authors, as this type of prediction uses graph information to generate predictions on nodal values through message passing rounds. In each message passing round, each node updates its nodal embedding using information aggregated from its neighbors' embeddings. This type of prediction integrates well with mesh-based simulations, such as VLM or RANS solvers used in this study, as the data structure of the simulation results can be easily represented by a graph G [57–59].

Recently, multiple studies have leveraged GNN architectures to predict physical fields, including across varying fidelities. Pfaff et al. proposed `MeshGraphNets`, which use message-passing mechanisms to learn mesh-based simulations on irregular and time-evolving computational grids with noticeable efficiency and accuracy for both static and dynamic systems [38]. This feature is desirable for the surrogate modeling task in this paper, as the surrogate model must not be sensitive to spatial discretization changes, such as mesh morphing introduced by geometry variation in the MDO cycle. Shao et al. proposed an auto-encoder GNN surrogate called `PIGNN-CFD` for rapid prediction of urban wind fields based on irregular unstructured mesh data from CFD simulations [39]. In their work, they added the Reynolds stress tensor residual to the loss function, thereby implicitly building physical constraints into the surrogate model. When applied to multi-scale and multi-fidelity applications, U-net [42] types of architectures are commonly used. Deshpande et al. Yang et al. and Wei and Nikolaos demonstrated the effectiveness of U-net type GNNs in physical simulations, particularly in the domains of finite element method (FEM)-based structural simulations and smooth particle hydrodynamics (SPH)-based fluid simulations [40,43,60]. Their work highlighted the superior performance of graph U-Nets in multi-scale representation learning, accommodating unstructured mesh data, modeling long-range interactions, and being mesh topology-agnostic. Finally, Liu et al. [41] used a graph attention network-based [61] surrogate for fluid simulations and found that the attention mechanism helps to describe physical relationships among graph nodes [41].

The multi-fidelity VortexNet model (`MF-VortexNet`), hereafter `VortexNet`, proposed in this paper, has an architecture inspired by the aforementioned works. Specifically, it utilizes a multi-head message aggregation and passing layer using Graph Attention Network v2 (`GATv2`) [62] and incorporates U-net-type skip connections between encoders and decoders [42]. The use of multi-head attention enables the network to attend to richer feature subspaces by concurrently building multiple regression models between a given node and its neighbors, thereby enhancing its feature-capturing capability. Additionally, as the network may require multiple layers of message aggregation and passing, U-net type skip connections are employed to short-circuit deep encoders, improving the network's efficiency in capturing low-frequency features. Overall, the network is designed to handle graph sizes typically used in VLM simulations, which contain hundreds to thousands of nodes. It aims to capture nonlinear flow patterns in RANS simulations and augment VLM results, providing a tool to increase design fidelity during the conceptual design phase.

Fig. 2 provides a high-level schematic of the proposed `VortexNet` network. In this figure, the computational graph before and after each major block is shown in black boxes. The graph consists of four nodes arranged in a Delta wing layout, and the five black solid lines connecting them represent the graph edges. The authors want to emphasize that the edges and nodes shown here are just for illustration purposes and do not represent the actual graph topology of the input data. The purple vectors (represented by purple boxes) beside each node are the corresponding node feature arrays, and the yellow vectors (represented by yellow boxes) beside each edge are the edge feature arrays. Specific details regarding the node and edge features are presented later in Table 2. As the graph passes through the network, the node feature array and edge feature array may change size, which is indicated by the lengthening or shortening of the boxes representing the arrays. Hence, each individual graph in the boxes represents a snapshot of the graph's state. In this chart, a black arrow indicates a direct connection among stages, and a blue arrow represents a U-net type skip connection that will be explained in the subsequent sections.

The input graph first goes through an Input Encoder, which maps the input features to a latent dimension. The encoded graph then goes through a series of Convolutional Blocks. These blocks use multi-head attention mechanisms, and hence the feature array size may be

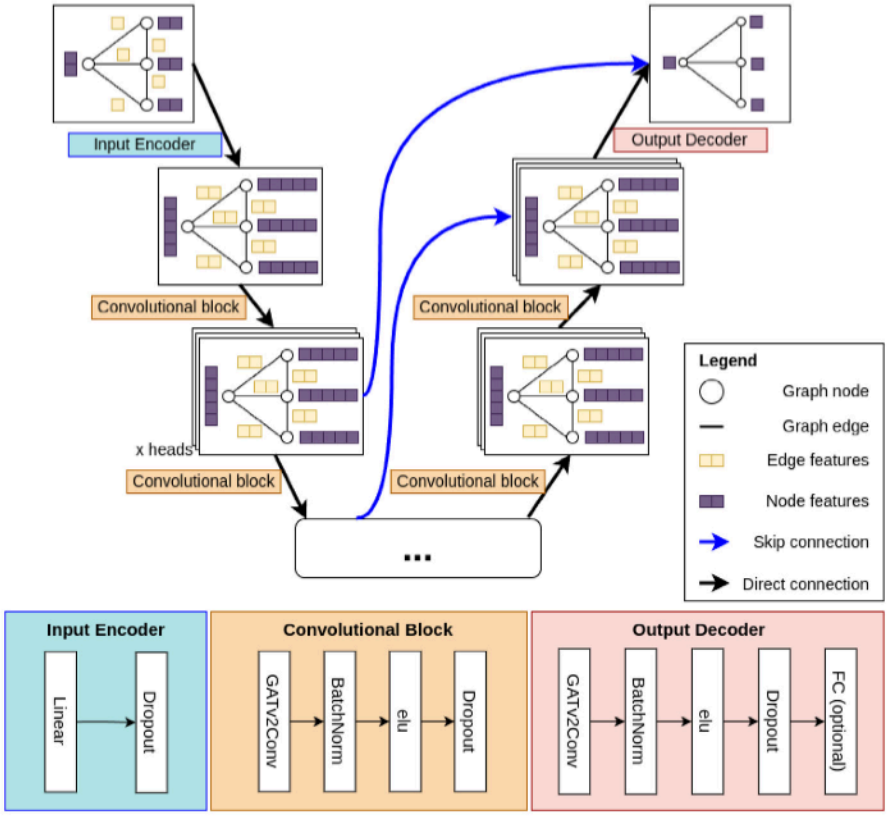


Fig. 2. Schematic of the proposed `VortexNet` GNN architecture. Colors indicate different network blocks. A black arrow represents direct message passing between blocks, and a blue arrow denotes a skip connection to the receiving block. The figure also presents snapshots of the graph at each computational step, showing nodes, edges, and their associated feature arrays.

multiplied by the number of heads. In this network, the first half of the Convolutional Blocks serves as the encoder, and the second half of the Convolutional Blocks serves as the decoder. The decoder blocks receive information from both their previous blocks and the graph information passed on by skip connections. Finally, an Output Decoder, which utilizes an architecture similar to the Convolutional Block but with an optional fully connected layer to adjust the output dimension, is used to produce nodal predictions for a field value of interest, such as pressure or velocity fields. The following sections describe the key components of the `VortexNet`.

2.4.1.1. Input encoders. The Input Encoder comprises two components: a node encoder and an edge encoder. The objective of including this input encoder is to uplift and enrich the input node features and edge features. In the current work, an 11-dimensional node features and a 1-dimensional edge features are used. Hence, a feature encoding to expand the input feature array to a higher-dimensional space is beneficial for the latter stage of the Convolutional Blocks, where latent relationships will be identified. A linear transformation is used for both the node encoder and the edge encoder, such that

$$\mathbf{h}_i' = \mathbf{W}_n \mathbf{h}_i + \mathbf{b}_n, \tag{13}$$

where $\mathbf{h}_i'$ is the encoded feature, $\mathbf{h}_i$ is the input feature, $\mathbf{W}_n$ are the learnable weights and $\mathbf{b}_n$ are the learnable biases. The node encoder encodes node feature to a dimension of $[\mathbb{F}\times\mathbb{F}]$, where $\mathbb{F}$ is defined by the hyper-parameter `hidden_feature`, and the edge encoder encodes edge feature to a dimension of $\mathbb{F}$. After the linear layer, a dropout layer is used for the node feature array.

2.4.1.2. Convolutional block. In each Convolutional Block, the primary mechanism for updating node features based on graph structure is by using `GATv2Conv` layers [62]. Specifically, the `GATv2Conv` computes a weighted average of the transformed featured of the neighborhood nodes for each nodes, expressed as,

$$\mathbf{h}_i' = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}\mathbf{h}_j\right), \tag{14}$$

where $\mathbf{h}_i'$ is the new feature representation at node i, $\mathcal{N}_i$ is the set of neighboring nodes connected to node i by direct edges, α_{ij} and $\mathbf{W}$ are the learned weights, $\mathbf{h}_j$ is the input feature array at node j, and σ is an activation function. The attention weights α_{ij} indicate the importance of node j's features to node i and allow every node to attend to every other node to which it is connected in the graph [62]. This mechanism enables the Convolution Block to capture how information propagates across the graph and, when the node and edge features are appropriately designed, to operate in a manner that is independent of the underlying mesh discretization. The actual attention function is defined as:

$$\alpha_{ij} = \mathrm{softmax}_j\left(e(\mathbf{h}_i, \mathbf{h}_j)\right), \tag{15}$$

and

$$e(\mathbf{h}_i, \mathbf{h}_j) = \mathbf{a}^T \mathrm{LeakyReLU}(W_n \cdot [\mathbf{h}_i \parallel \mathbf{h}_j] + W_e \cdot E_{i,j}), \tag{16}$$

where $e(\mathbf{h}_i, \mathbf{h}_j)$ is the feature importance score of neighbor node j to node i. Matrices $\mathbf{a}$, W_n and W_e are learned weights, $E_{i,j}$ are the corresponding edge features, and $(\parallel)$ denotes matrix concatenation [62].

The `PyTorch GATv2Conv` layer employs multiple attention heads that allow the model to attend to information from different representational subspaces. In the multi-head attention setting, multiple independent attention mechanisms execute the transformation of Eq. (14), and the resulting feature vectors $\mathbf{h}_i'$ are then concatenated.

Utilizing this `GATv2Conv` layer, the Convolution Block consists of one `GATv2Conv` layer that maps the input graph to an output graph

with a node feature array in $\mathbb{R}^{\mathrm{F}\times\mathtt{heads}}$ after concatenation. Additionally, `heads` is a hyper-parameter specifying the number of heads. The input–output dimension is consistent across all layers except for the first Convolution Block after the Input Encoder, where the input feature dimension is $\mathbb{R}^{\mathrm{F}\times\mathrm{F}}$. After the `GATv2Conv` layer, the feature passes through batch normalization, an exponential linear unit (elu) activation function, and a dropout regularization layer. The dropout layer includes a tunable hyper-parameter named `dropout_rate`.

In the `VortexNet` architecture, the Convolutional Block is repeated `hops` − 1 times, where `hops` is a tunable hyperparameter specifying the network depth. It is worth noting that in many previous graph U-net-like architectures, pooling layers are used to successively reduce the computational graph size in deeper encoder layers to minimize non-influential information overhead [40,43,59,60]. However, such pooling layers are not used in the current network because the implications of using these methods under the current MF learning setting remain uncertain to the authors at this time: although sequential graph sparsification across layers improves encoding efficiency, the final field prediction must ultimately be generated at the original training targets' resolution. Consequently, unpooling operations are required to reconstruct a dense graph from the coarsened latent representation [59]. These reconstruction procedures may introduce systematic bias into the predicted fields and therefore require careful architectural design and model validation. In contrast, architectures without pooling sacrifice computational efficiency but avoid the potential reconstruction-induced bias. This trade-off led the authors to omit nodal pooling between layers. A discussion on the potential future direction of including pooling layers in the `VortexNet` architecture is presented in the Results Section 4.

2.4.1.3. Output decoder. The Output Decoder's major role is to map latent space features to the desired output dimension. This block contains layers identical to those used in a Convolution Block, except that the `GATv2Conv` layer now maps the feature array from $\mathbb{R}^{\mathrm{F}\times\mathtt{heads}}$ to $\mathbb{R}^{1}$. An optional fully connected (FC) layer is appended at the end of this block to enable graph size adjustment for the final output. In the context of this work, since the input graph and training labels have identical graph sizes, this fully connected layer is not used.

Although edge features can be extracted after the Output Decoder block, they are discarded. Consequently, the Output Decoder provides scalar value predictions at each node, which represent the field value predictions that mimic the HF training labels.

2.4.1.4. Skip connections. The skip connection is marked by the blue curved arrow in Fig. 2. For the ith Convolutional Block, the output is connected to the (`hops` − i)th block's output. The first encoder block is skip-connected to the output decoder, the second to the final decoder block, and so on in reverse order. Dimension adjustment is required for the first skip connection only, through a linear layer; all other encoder–decoder pairs have matching feature dimensions and therefore require no adjustment. A skip connection weight hyper-parameter (`Alpha`) is used to control the amount of skip connection information that the subsequent decoder utilizes, such that the feature array $\mathbf{h}_i$ seen by the subsequent decoder is

$$\mathbf{h}_i = \mathrm{A}\mathbf{h}_{i,\text{skip connection}} + (1-\mathrm{A})\mathbf{h}_{i,\text{previous block}}, \tag{17}$$

and this skip connection weight (A) is applied uniformly across all skip connections.

2.4.2. Training

The prediction accuracy of `VortexNet` relies heavily on the training scheme. Throughout experiments, it has been observed that a training scheme with a combination of learning rate scheduling, regularization techniques, and physical loss terms helps `VortexNet` to achieve higher prediction accuracy. The following sections discuss the training scheme used in this work.

2.4.2.1. K-fold training. The entire dataset, as described in Section 3.1, is randomly partitioned into a 30%–70% test-training split. For network training, only the training set is exposed to the `VortexNet`, while the test set is reserved for performance characterization. The training set is further randomly partitioned into four batches. In each fold of the k-fold training, one batch of the training set serves as the cross-validation set while the remaining subsets serve as the training set. This technique exposes the model to diverse data distributions and prevents overfitting to specific cross-validation subsets [63].

In standard k-fold cross-validation training, the models trained in each fold are independent and the goal is to evaluate the robustness of the architectural design. If there is a need to select a model, a voting algorithm is typically employed. In the current study, however, we made minor adjustments to the model voting mechanism. Instead of training mutually independent models in each fold, our training script recycles the best model weights identified so far from all previous folds' training. Specifically, a register actively stores the best-performing model, as indicated by the cross-validation loss, and in each fold training resumes from the current best model.

In using this training scheme, the cross-validation set in the later folds may overlap with the training data of previous folds, potentially inflating the model performance in later folds. However, this strategy was chosen for two reasons. First, the size of the total training set is limited compared to the number of trainable weights in the `VortexNet`. Using standard k-fold training prohibits making full use of the training set. Second, the way physical loss terms are used in training necessitates such model recycling, as discussed in the next section.

2.4.2.2. Loss function. The loss function used for training is a compound loss defined as:

$$\mathcal{L} = \mathrm{SmoothL1Loss}\left(q^{HF}, q^{\mathrm{pred}}\right) + \Lambda\,\mathrm{PhysicalLoss}\left(q^{\mathrm{pred}}\right), \tag{18}$$

where "SmoothL1Loss", and "PhysicalLoss" are the loss components; q^{HF} is the high-fidelity data provided as training labels; $q^{\mathrm{pred}} = \mathcal{F}\left(q^{\mathrm{LF}}, \omega_{\mathcal{L}}\right)$ is the `VortexNet` predicted field output, and Λ is the weight of the physical loss which is controlled by two hyper-parameters as described in Eq. (20). The "SmoothL1Loss" is well-documented in the `PyTorch` manual [64].

The "PhysicalLoss" term is introduced to improve the model's prediction capability. In designing surrogate models, two typical approaches are commonly employed: physics-driven and data-driven. In the physics-driven approach, the governing equations can either be embedded in the aggregation function or incorporated into the loss term. For surrogate models aimed at HF flow field prediction, the residuals of the Navier–Stokes equations at selected points are often used to drive the network convergence [65]. While accurate, such networks are computationally expensive to train. Alternatively, in the data-driven approach, the physical loss is omitted. However, this approach often suffers from high variance in prediction errors.

Considering that the proposed surrogate model is intended for conceptual design purposes, where trend-wise accuracy and delta values of QOIs are more critical than the absolute prediction accuracy, we aim to employ a lightweight physical constraint. This constraint helps calibrate the surrogate model predictions for trend-wise quality without adding significant complexity to the network architecture. In this study, we propose a physical loss term based on LF VLM simulation to improve the network's trend-wise prediction quality. Variations in nonlinear flow features, such as boundary layer effects, could be considered to be equivalent to the errors committed through violations of the "PhysicalLoss" term.

To compute the physical loss, recall that VLM solves the linear equation shown in Eq. (4). When a surrogate model is used to predict $q^{\mathrm{pred}} = \Delta C_p^{\mathrm{pred}}$, this field can be passed backward through the stored lattice geometry information to compute the predicted vortex strength distribution, Γ^{pred}. The physical loss term can thus be expressed as a 2-norm of the residual:

$$\mathrm{PhysicalLoss}(q^{\mathrm{pred}}) = ||AIC \cdot \Gamma^{\mathrm{pred}}(q^{\mathrm{pred}}) - RHS||. \tag{19}$$

In the above physical loss computation, no real-time computation of the AIC and RHS matrices is required, as this information can be stored during the dataset assembly phase, thereby creating minimal overhead for network training.

When the physical loss weight Λ is large, the surrogate-predicted ΔC_p^{pred} will closely resemble the VLM results. However, an excessively large $Lambda$ prevents the surrogate from learning the LF-to-HF correction. To address this issue, an exponential decay of the physical loss weight Λ is applied during the k-fold training. Initially, the model is trained to closely match the VLM solver, so that a good trend-wise baseline is established. While the model keeps the learned mapping, the physical constraint is gradually relaxed. Towards the final folds, the surrogate model's training convergence is primarily driven by data to capture the non-linear flow patterns present in HF simulations. The following equation is used to decay the physical loss weight across folds:

$$\Lambda_i = \Lambda^{\max}\left(\Lambda^{\min} + \left(1-\Lambda^{\min}\right)\exp\left(-\ln\left(\frac{1}{\Lambda^{\min}}\right)\frac{i-1}{k-1}\right)\right), \quad \text{for } i \in \{1,\ldots,k\}, \tag{20}$$

where $\Lambda^{\max}$ is a hyper-parameter (`max_phys_loss`) representing the initial physical loss weight; $\Lambda^{\min}$ is a hyper-parameter (`min_phys_loss`) that specifies the target final magnitude of the physical loss weight; and k is the number of folds in k-fold training.

2.4.2.3. Optimizer and training scheme. Optimization is performed using the ADAM optimizer with an initial learning rate specified by the hyperparameter `learning_rate` and a decay rate defined by `decay_rate` [66]. A dynamic learning rate scheduler is implemented to halve the learning rate when the validation loss ceases to improve across training epochs. In addition, an early stopping criterion is applied: if model performance does not improve for 150 consecutive epochs, training for the current fold is terminated and the best-performing model up to that point is retained to mitigate overfitting. The validation loss extends the objective defined in Eq. (18) by incorporating a sign-violation regularization term. This term is defined as the average number of nodal prediction sign violations relative to the ground truth. Including this indicator in the validation metric supports more reliable early stopping by discouraging physically inconsistent predictions. In particular, it reduces overfitting that may otherwise introduce high-variance errors in the afterbody wing panels, where ΔC_p magnitudes are typically small but contribute significantly to aerodynamic moment due to their relatively large moment arms. To further improve prediction robustness, 3% random noise is added to all node features during training. Across all experiments and hyperparameter configurations, a batch size of 32 is used. Training is conducted on an NVIDIA A100 graphics processing unit (GPU).

2.4.2.4. Hyper-parameter optimization. The surrogate model possesses nine hyper-parameters that need to be optimized. The hyper-parameter optimization study is conducted using `Optuna`, an open-source hyper-parameter optimization framework that automates hyper-parameter search using advanced sampling schemes and efficient pruning algorithms [67]. The hyper-parameters, along with their bounds listed in Table 1, define the search space, and `Optuna` automatically identifies the optimal hyper-parameter settings that minimize the model test loss over the test set. The model test loss is defined as the Mean Squared Error (MSE) between the predicted ΔC_p^{pred} and the corresponding HF ΔC_p^{HF} across the test set.

In total, 80 hyper-parameter configurations are generated by `Optuna` in this study. The training time varies across configurations, but on average, each configuration takes 1.75 h to train. The average test loss across all hyper-parameter configurations is 4.13×10^{-2}. Across trained models with test losses below 3×10^{-2}, the aerodynamic coefficients computed from the predicted fields are similar. However, the predicted pressure fields exhibit minor stylistic differences, characterized by localized, small-area checkerboard-like fluctuations. These patterns are low in amplitude and do not correspond to large-magnitude oscillations. The final model is selected for its robustness, exhibiting consistent field prediction accuracy across the design space.

Table 1
List of the hyper-parameters studied, along with their value range and definitions.

Hyper-parameter name	Value range	Definitions
`hops`	int[2, 16]	total layers of Convolutional Block
`hidden_feature`	int[3, 64]	hidden layers feature width
`heads`	int[1, 8]	number of heads for multi-heads attention
`dropout_rate`	[0.05, 0.4]	dropout regularization weight
`Alpha`	[0.0, 1.0]	skip connection bandwidth
`max_phys_loss`	[0, 0.5]	maximum physical loss weight in Eq. (20)
`min_phys_loss`	$[1, 1000] \times 10^{-5}$	target physical loss weight in Eq. (20)
`learning_rate`	$[1, 1000] \times 10^{-4}$	initial learning rate to the ADAM optimizer
`decay_rate`	$[1, 10000] \times 10^{-6}$	decay rate to the ADAM optimizer

The final selected model adopts nine Convolution Blocks with an additional output decoder, resulting in a total of ten message-passing rounds. The hidden feature dimension is 63 with four attention heads. A dropout rate of 8.5×10^{-2} is applied. The skip-connection bandwidth is 9.7×10^{-3}, indicating that most information propagates through the deeper layers. The initial physics loss is 0.42 and decays to 5.8×10^{-5} throughout training. The learning rate is 1.3×10^{-4}, with a decay rate of 1.6×10^{-6}.

3. Dataset

3.1. Dataset overview

The dataset for surrogate model training and testing consists of 15 3-D Delta wing geometries at two levels of fidelity. The HF surface pressure field data is generated using the SU2 CFD solver as described in Section 2.2. The flow domain is discretized using `Pointwise`'s automatic unstructured volume mesh capability [68]. The outer mold line of each underlying geometry is accurately captured by enforcing a maximum surface mesh-to-geometry deviation of less than 1×10^{-5} [m]. Across all geometries, identical mesh property definitions are used to ensure consistency in global mesh properties and boundary conditions, including the T-Rex boundary layer mesh growth profile, leveraging `Pointwise`'s scripting capability. Due to variations in surface body geometry, the total number of elements in the fluid domain ranges from 4.89 to 19.47 million. The resultant mesh achieves $y^+ < 1$ for most parts of each wing, except near the wing tips at the trailing edge, satisfying the turbulence model's y^+ requirements.

The LF data is generated using the VLM implementation in SUAVE. For all geometries, a 30 × 30 lattice discretization is applied in the chord-wise and span-wise directions. The lattice is evenly distributed in the chord-wise direction and follows a cosine distribution in the span-wise direction.

The 15 Delta wing geometries are parameterized using two design variables: leading edge sweep angle and root airfoil shape. The leading edge sweep angles used in this study are 55°, 65°, and 75°. Across all geometries, the root chord length is kept constant at 0.65 [m], resulting in variations in span and reference area of the Delta wing. The root airfoil shapes are taken from NACA 4-digit airfoils, and in this study, NACA 0010, NACA 0016, NACA 0024, NACA 2416, and NACA 4416 are used. All 15 combinations of the root airfoil shape and leading edge sweep angle are generated using SUAVE.

Each geometry is then simulated under 40 different free-stream conditions, varying AOA, Mach number (Ma), and Reynolds number

(Re). The AOA is sampled from 0° to 20°, the Ma from 0.35 to 0.5, and the Re from 6.5×10^6 to 1×10^7. A Latin Hypercube Sampling (LHS) technique is used to generate samples across the three free-stream variables. It is worth noting that the LF VLM model does not utilize the Reynolds number in its simulation. In total, 600 test samples are generated across the geometry design space and the free-stream variable space.

Two additional holdout datasets are prepared using a similar workflow to assess the model's generalizability to previously unseen geometries and free-stream conditions. These datasets are introduced in Sections 4.1.2 and 4.2.1, respectively, and are not exposed to the surrogate model during training. All CFD runs are conducted on the Sherlock Cluster at the Stanford Research Computing Center (SRCC), utilizing a 64-core 2.4 GHz partition.

3.2. Dataset preparation

The HF and LF datasets differ significantly in their dimensions. While the LF data resides in $\mathbb{R}^{30\times30}$, the HF data dimension is defined by the number of surface mesh nodes, which varies based on the geometry. Although GNN-based architectures can accommodate mapping between these different graph structures, training such surrogate models can be computationally expensive and is not the focus of this work.

In this study, the goal is to demonstrate the capability of utilizing such architectures to build surrogate models that enhance conceptual vehicle design at a relatively low computational cost. To achieve this, the authors decided to standardize the HF and LF graphs by projecting data from both sources onto a reference state. One approach is to use the lattice panels defined in the VLM as the reference plane and project the HF surface pressure field onto this surface. As VLM computes the local loading coefficient $\Delta C_p^{\mathrm{LF}} = C_{p,\mathrm{lower}} - C_{p,\mathrm{upper}}$, it was decided that the standardization step for the HF data should project the surface pressure onto the lattice panels to generate the HF local loading coefficient, ΔC_p^{HF}. This process is described by the following equations:

$$\Delta C_p^{\mathrm{HF}} = C_{p,\mathrm{lower}}^{\mathrm{HF}} - C_{p,\mathrm{upper}}^{\mathrm{HF}}, \tag{21}$$

where C_p^{HF} is a vector in $\mathbb{R}^{30\times30}$. The pressure coefficient for each side can be calculated by:

$$C_{p,\mathrm{side}}^{\mathrm{HF}} = \frac{1}{q_{\mathrm{ref}}}\left[\frac{\sum_{i\in\mathcal{I}}\delta_{\mathrm{side}}(i)P_iA_i}{\sum_{i\in\mathcal{I}}\delta_{\mathrm{side}}(i)A_i}\right]_{\mathcal{I}=1}^{N}, \tag{22}$$

where P_i is the pressure, A_i is the cell area at HF node i, $\mathcal{I}$ is the set of all surface nodes whose (x_i, y_i) coordinates lie within the boundary of a lattice panel, indexed from 1 to $N = 30 \times 30$, under vertical projection, and q_{ref} is the reference pressure computed from free-stream conditions. $\delta_{\mathrm{side}}(i)$ is an upper or lower surface selection mask that identifies whether a node belongs to the lower or upper surface based on its z_i coordinate (applicable for the root airfoils used in the current study). For example, the $\delta_{\mathrm{lower}}(i)$ mask can be defined as:

$$\delta_{\mathrm{lower}}(i) = \begin{cases} 1 & \text{if } z_i < 0, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

3.3. Node and edge features

As discussed in Section 3.1, the VLM simulation is Reynolds number agnostic. To enrich the aerodynamic features used in the surrogate model, free-stream conditions, including free-stream AOA, Ma, and Re, are incorporated alongside the local loading coefficient ΔC_p^{LF}. Because the proposed surrogate model is intended to augment aerodynamic prediction for design space exploration in conceptual design, it is essential to include geometric properties that influence aerodynamics in the surrogate model inputs. These geometric properties include wing thickness, surface curvatures, normalized lattice panel control point coordinates, and surface slopes. All geometric features are defined and evaluated at the nodal level. The control point coordinates are normalized using a characteristic length. In this study, the root chord length of 0.654 [m] is selected because it remains invariant across geometries within the design space. Furthermore, all geometries are translated so that the wing apex coincides with the origin. This treatment facilitates scale-invariant surrogate modeling and improves model generalizability.

Table 2
List of the nodal and edge features used by the `VortexNet`, along with their dimension and intended purposes.

Feature name	Dimension	Purpose
ΔC_p^{LF}	$[N_{\mathrm{nodes}} \times 1]$	VLM local loading coefficient
AOA $[rad]$	$[N_{\mathrm{nodes}} \times 1]$	free-stream AOA
Ma_∞	$[N_{\mathrm{nodes}} \times 1]$	free-stream Mach number
Re_∞	$[N_{\mathrm{nodes}} \times 1]$	free-stream Reynolds number
Thickness	$[N_{\mathrm{nodes}} \times 1]$	wing thickness at lattice panel control points
Upper Surface Curvature	$[N_{\mathrm{nodes}} \times 1]$	wing upper surface curvature at lattice panel control points
Lower Surface Curvature	$[N_{\mathrm{nodes}} \times 1]$	wing lower surface curvature at lattice panel control points
Upper surface slope	$[N_{\mathrm{nodes}} \times 1]$	wing upper surface chord-wise slope at lattice panel control points
Lower surface slope	$[N_{\mathrm{nodes}} \times 1]$	wing lower surface chord-wise slope at lattice panel control points
x_c	$[N_{\mathrm{nodes}} \times 1]$	normalized lattice panel control points x-coordinates
y_c	$[N_{\mathrm{nodes}} \times 1]$	normalized lattice panel control points y-coordinates
Euclidean distance	$[4N_{\mathrm{nodes}} \times 1]$	normalized Euclidean distances from one control point to its four nearest neighbors'

At the edge level, the Euclidean distance computed from normalized coordinates between nodes is used to represent nodal connectivity patterns and spatial relationships among nodes. For each node, the Euclidean distances to its four nearest neighboring nodes are included in the edge feature. The combination of nodal and edge features allows the `VortexNet` surrogate model to capture the spatial and aerodynamical information for aerodynamic characterization.

The nodal and edge features used by the `VortexNet` surrogate model, along with their intended purposes, are summarized in Table 2.

Once these node and edge features are assembled, a z-score standardization is performed for all nodal features except for the normalized coordinates, as they are scaled by the characteristic length. After these dataset preparation steps, the LF graph and HF training label can be fed into the surrogate for training.

4. Results

In this work, we focus on predicting local loading coefficients (ΔC_p). Once ΔC_p is predicted, arbitrary QOIs can then be calculated using the existing multi-fidelity workflow as discussed in Section 2.3. To highlight the effectiveness of the proposed multi-fidelity workflow, two prediction accuracy tests are performed. First, we evaluate the surrogate model `VortexNet`'s ΔC_p prediction quality. We examine the root-mean-square error (RMSE) of panel-wise $\Delta C_p^{\mathrm{pred}}$ prediction against the projected CFD reference value ΔC_p^{HF} for geometries within the surrogate model's training set across various free-stream conditions. To further demonstrate `VortexNet`'s generalizability in maintaining comparable prediction quality across geometries within the dataset's design space, unseen geometries are introduced and the surrogate model's accuracy is evaluated. These results are presented in Section 4.1. Second, to quantify the aerodynamic QOI prediction accuracy gain of adopting the multi-fidelity framework, lift (C_L) and pitching moment (C_M) coefficients are evaluated using `VortexNet`-predicted $\Delta C_p^{\mathrm{pred}}$ and compared against the CFD reference and VLM benchmark.

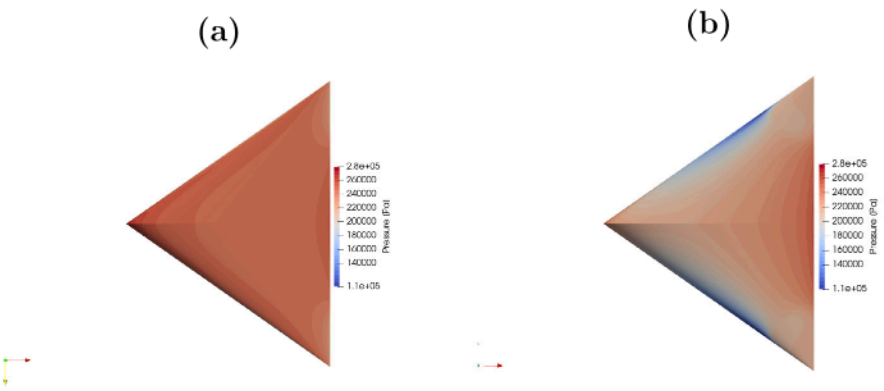


Fig. 3. The HF surface pressure at the lower (figure (a)) and upper (figure (b)) surfaces obtained from CFD for a wing with 55° leading edge sweep and a root airfoil profile of NACA 2416 at AOA= 16.4°, $M_a = 0.47$, and $R_e = 9.78 \times 10^6$.

This comparison is conducted for geometries that are used for `VortexNet` training, and for the unseen geometries as mentioned in Section 4.1. The results are presented in Section 4.2. Overall, the proposed multi-fidelity workflow showed robustness with respect to underlying geometry changes and a significant fidelity gain over the VLM benchmark. To rationalize and understand the prediction mechanism and assist further development of `VortexNet`-like GNN-based surrogate models, a latent space ablation study is presented in Section 4.3 to discuss how the graph features are utilized across the latent layers in `VortexNet`.

4.1. Local loading coefficient (ΔC_p) predictions

To evaluate the model's performance, we consider two distinct aspects. First, to quantify its prediction accuracy, we examine the RMSE of the field predictions ΔC_p^{pred} against the projected CFD references ΔC_p^{HF}, as outlined in Section 3.1. Second, to assess the model's generalizability and its potential for assisting in design space exploration, we evaluate its ΔC_p^{pred} RMSE against ΔC_p^{HF} using new geometries that were held out from the surrogate model training. The following sections discuss these two assessments.

4.1.1. Prediction for seen geometries during training

The RMSEs for the training set, test set, and the entire dataset are 0.1064 with an R^2 value of 0.9519, 0.1065 with an R^2 value of 0.9500, and 0.1064 with an R^2 value of 0.9513, respectively. No significant difference between the test and training sets is observed, and most of the variance in the dataset is captured by the trained model, indicating that the resultant model is well-trained.

To visualize the ΔC_p from the VLM, `VortexNet` predictions, and CFD references, a geometry configuration at high AOA was randomly selected, featuring a leading edge sweep angle of 55° and a root airfoil of NACA2416. The free-stream conditions are an AOA of 16.4°, Ma of 0.47, and a Re of 9.78×10^6. The corresponding CFD pressure on the upper and lower wing surfaces are presented in Fig. 3. As shown in the figure, the surface pressure exhibits a "vortex lift" pattern: on the upper surface toward the wing outboard near the trailing edge, suction effects induced by the separated vortex are evident.

The aforementioned surface pressure is then normalized by the corresponding reference pressure and projected onto lattice panels. Fig. 4(c) shows the projected ΔC_p^{HF}, which serves as the "ground-truth" reference. The projected pressure field preserves flow features from the CFD, including faint traces of vortex lift at the wing outboards. In Fig. 4(a), the ΔC_p^{LF} computed from the VLM is shown. This pressure field follows potential flow theory, and no vortex lift or flow separation is observed. The VLM pressure field serves as input to `VortexNet` as part of the nodal features.

Table 3
Test set RMSEs between ΔC_p^{pred} and ΔC_p^{HF}, grouped by wing geometries.

Sweep angle [°]	Root airfoil (NACA 4-digits)				
	0010	0016	0024	2416	4416
55	0.0566	0.0964	0.1330	0.0872	0.0875
65	0.0942	0.1159	0.0577	0.1046	0.1016
75	0.1579	0.0945	0.0900	0.1506	0.1286

Fig. 4(b) shows the ΔC_p^{pred} predicted by `VortexNet`. Comparing these pressure fields, it is evident that the predicted pressure field exhibits strong visual similarity to the projected CFD pressure field, both in the predicted ΔC_p values and in the overall pressure distribution. However, artifacts of reconstruction error are noticeable near the wing apex at the central body. Despite these discrepancies, the authors believe that the surrogate model's predictions provide higher fidelity than those of VLM, as it captures nonlinear flow features.

Although not shown, ΔC_p^{pred} predictions for other geometries and free-stream conditions follow a similar pattern, with `VortexNet`'s predictions appearing visually closer to the projected CFD fields (ΔC_p^{HF}) than to those of the VLM (ΔC_p^{LF}).

To provide a more quantitative analysis of the ΔC_p prediction error, we compared the RMSE across all geometries in the test set. The corresponding values are presented in Table 3. The RMSE ranges from 0.0566 to 0.1579 in the observed dataset with a standard deviation of 0.0366 across all geometries. Using the leave-one-out one sample t-test, under the null hypothesis that the prediction RMSEs are consistent for all geometries, it is observed that one geometry with a leading-edge sweep angle of 75° and root airfoil shape of and NACA 0010 is an outlier. For the remaining geometries, the trained model maintains a relatively consistent prediction accuracy. Overall, `VortexNet` has maintained generalizable field prediction capability across the design space. However, the performance discrepancy among geometries indicates that validating model generalizability is necessary before deploying surrogate models to assess whether prediction accuracy across the design space is sufficient for downstream applications.

4.1.2. Prediction for unseen geometries

One potential use case of the proposed `VortexNet` is to assist in design space exploration during conceptual design. In such applications, a design space is first defined by specifying a set of geometric parameters and their respective bounds. A coarse sampling of geometries, using both HF and LF methods, is then conducted to train the surrogate model. Subsequent design space exploration can rely on the surrogate model's augmentation of LF methods to provide quasi-high-fidelity aerodynamic evaluations. Consequently, it is essential to

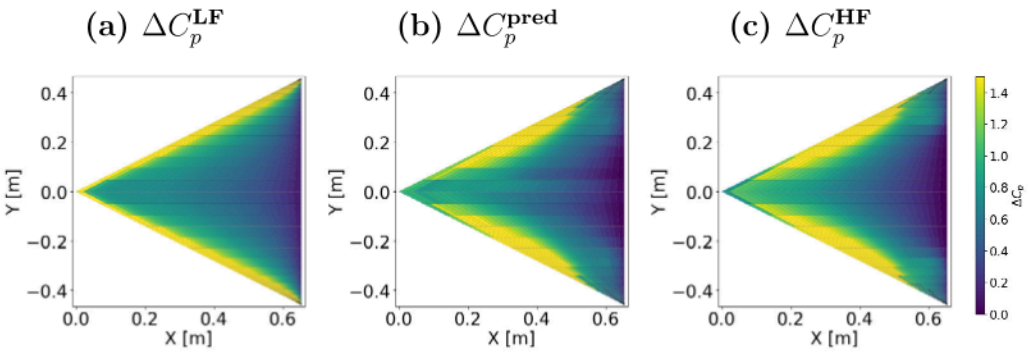


Fig. 4. Lattice panels and local loading coefficients ΔC_p are visualized using color maps ranging from $\Delta C_p = 0$ to $\Delta C_p = 1.5$, with (a) VLM ΔC_p^{LF}, also serving as the LF input, (b) `VortexNet` $\Delta C_p^{\mathrm{pred}}$, and (c) the reference HF ΔC_p^{HF} obtained by projecting the CFD surface pressure onto the lattice panels.

Table 4
Unseen geometries RMSEs.

Sweep Angle [°]	Root airfoil (NACA 4-digits)	
	0013	3416
60	0.1441	0.2695
70	0.2106	0.3293

evaluate the proposed surrogate model's generalizability to geometries not included in the training dataset.

To study generalizability, we tested geometries and feature arrays not encountered by the network during training and evaluation. Four design points within the geometric parameter ranges of the training set's design space, as tabulated in Table 3, were selected. Specifically, we chose two leading-edge sweep angles (60° and 70°) and two root airfoils (NACA 0013 and NACA 3416) that induce thickness and camber variations. A combination of these design variables yields four new wing geometries: 60° NACA 0013, 60° NACA 3416, 70° NACA 0013, and 70° NACA 3416. Each geometry is then simulated under 11 different far-field conditions, with AOA, Mach number, and Reynolds number sampled using LHS. The sampling bounds for AOA, Ma, and Re are $[0°, 20°]$, $[0.35, 0.5]$, and $[6.5, 10]\times10^6$, respectively, matching those used to construct the training set as discussed in Section 3.

The resulting RMSEs are listed in Table 4. The maximum RMSE observed is 0.3293 for the 70° Delta wing with the NACA 3416 airfoil, and the minimum RMSE is 0.1441 for the 60° Delta wing with the NACA 0013 airfoil. On average, the RMSE is 0.2481, representing a 132.9% increase from the RMSE reported in the previous tests on seen geometries. A reduction in prediction accuracy for unseen geometries is expected, as a similar trend is reported in the previous study [44].

To better understand the potential origin of this degradation, we examine how the RMSE varies with the type of geometric variation from the training set. Within this limited new-geometry dataset, the errors appear to correlate more strongly with camber variation. Configurations introducing a new camber shape exhibit larger RMSE. A hypothesis is that camber variation modifies the global pressure loading pattern and redistributes ΔC_p over a larger portion of the surface, thereby altering both magnitude and spatial gradients of the field. As the current training dataset exposes `VortexNet` to a limited representation of camber profiles, the model's performance may be more sensitive to camber variation. Expanding the design space coverage to include more camber shapes in the `VortexNet` training set may help reduce prediction errors.

To further diagnose the source of error, a qualitative study was conducted by visually examining the resulting $\Delta C_p^{\mathrm{pred}}$ fields with reference to VLM and CFD ΔC_p results. For comparison across geometries, we present the ΔC_p for all geometries at an AOA of 18°, Ma of 0.4 and Re of 7×10^6. The results are presented in Fig. 5.

When comparing `VortexNet`'s prediction accuracy to the results shown in Fig. 4, we observe that for previously unseen geometries, the surrogate model introduces more artifacts in its predictions. For the 70° NACA 0013 configuration, the surrogate model fails to accurately capture the highest ΔC_p values. Small-wavelength artifacts of high- and low-pressure variation can be seen for 70° geometries near the leading edges toward the wing tips. These checkerboard-like pressure fluctuation patterns are most likely to be the dominant contributing factor for increased RMSE for these test geometries, suggesting that the dominant error mode is not a global bias in pressure magnitude but rather a high-frequency spatial inconsistency. Further examination in Section 4.2.2, which evaluates the accuracy of integrated QOIs, further supports this interpretation.

Despite these artifacts, when considered alongside other fidelity methods, `VortexNet` continues to demonstrate an intermediate level of capability between VLM and CFD in capturing major flow features. The authors believe that the model's generalizability and prediction accuracy can be further enhanced by refining the surrogate model architecture to promote spatial smoothness and multi-scale consistency, as discussed in Section 4.3, and byimproving the training data sampling scheme.

4.2. Aerodynamic coefficients predictions

We consider the CFD-predicted surface pressure coefficients and aerodynamic coefficients as the "ground-truth" reference. The error is thus the difference between QOIs computed with and without using the surrogate model. In the current study, two sources of error exist. The first is an epistemic discretization error introduced during the surrogate model training database preparation. Instead of using CFD surface pressures (C_p), the surrogate model is trained on and predicts projected CFD surface pressures ΔC_p^{HF}. This projection process introduces discretization errors that cannot be reduced during model training and will lead to differences in computed QOIs. The second error is the surrogate model's modeling error due to its imprecise representation of ΔC_p^{HF}. This error can be reduced through model training and hyperparameter optimization. In this study, the surrogate's modeling error is minimized by using optimal model hyper-parameters, while we have not extensively explored error reduction techniques for the epistemic discretization error.

We assess the workflow's performance from two perspectives. First, to evaluate the fidelity gain of adopting `VortexNet` in the multi-fidelity workflow, we study the QOIs performance of the workflow on geometries seen during training but at unseen free-stream conditions. We compare the QOIs accuracy obtained from the VLM, the `VortexNet` surrogate model, and projected CFD against the CFD ground truth. These results are presented in Section 4.2.1. Second, we examine the QOIs prediction characteristics of the proposed workflow on geometries not seen during surrogate model training, to demonstrate its performance in interpolating within the design space. This generalization assessment is presented in Section 4.2.2.

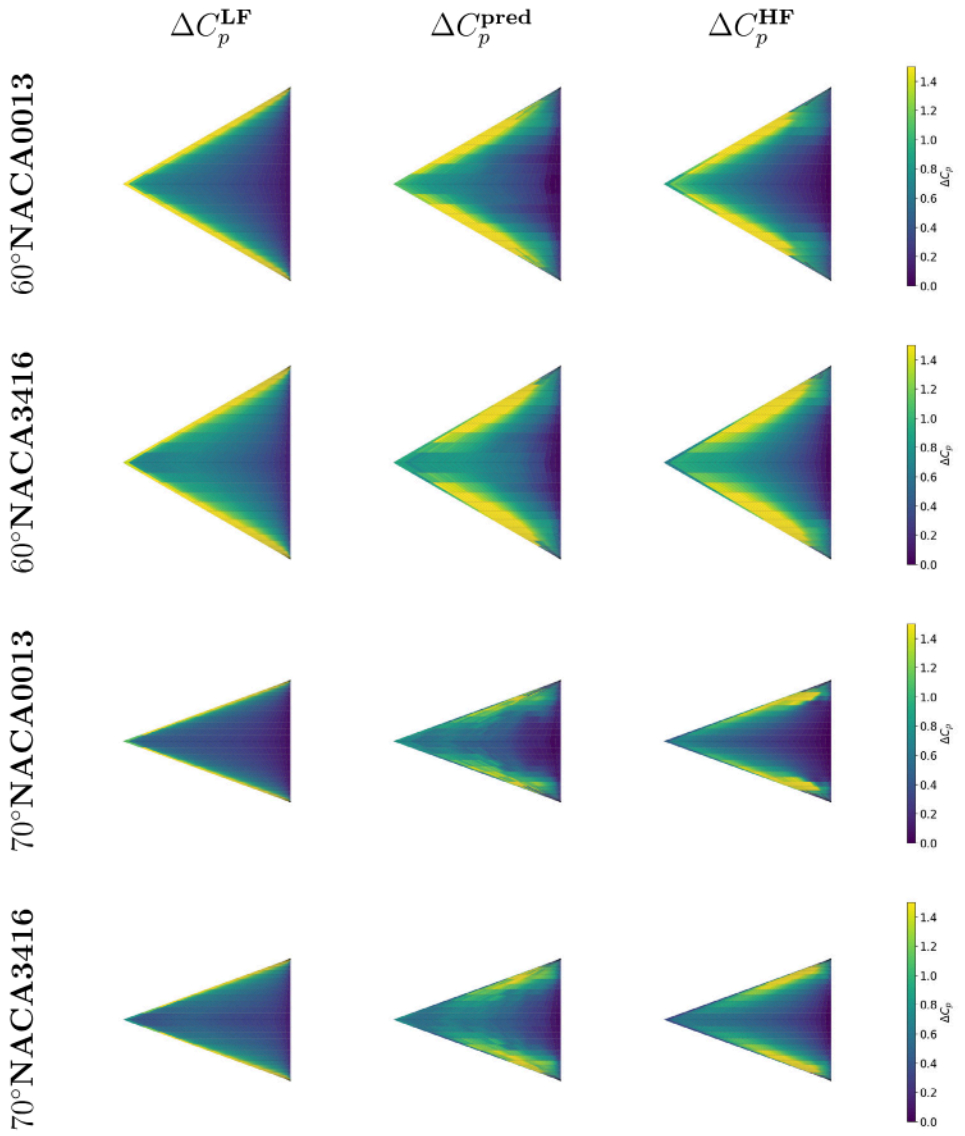


Fig. 5. Lattice panels and local loading coefficients ΔC_p are visualized using color maps ranging from $\Delta C_p = 0$ to $\Delta C_p = 1.5$ for four new geometries. From top to bottom: 60° NACA 0013, 60° NACA 3416, 70° NACA 0013, and 70° NACA 3416. From left to right: ΔC_p^{LF} obtained from the VLM, `VortexNet` $\Delta C_p^{\mathrm{pred}}$ predictions, and the projected CFD ΔC_p^{HF}.

4.2.1. *Aerodynamic coefficients prediction for seen geometries*

Our first effort to quantify the proposed workflow's QOIs prediction accuracy compares QOIs integrated from the surrogate model predictions $\Delta C_p^{\mathrm{pred}}$ with those integrated from the projected CFD surface pressures ΔC_p^{HF}, which represent `VortexNet`'s modeling error, and QOIs obtained directly from CFD, which contain both the epistemic discretization error and the surrogate model's modeling error. These QOIs are compared against the results computed directly by VLM, which serves as the LF benchmark. We examine the C_L and C_M polars over an AOA range from 5° to 25° in 2° increments. The interval from 20° to 25° lies outside the training set range (0°–20°; see Section 3) and is intentionally included to assess model robustness under mild extrapolation. Although the drag coefficient can be computed using VLM, the absence of viscous effect modeling renders cross-comparisons between VLM and CFD misleading. Therefore, this study focuses exclusively on lift and moment coefficient evaluations. The free-stream Mach number is held constant at 0.4, and the Reynolds number at 7.0×10^6.

Fig. 6 shows the relative error for C_L and C_M with AOA obtained from VLM for each individual Delta wing geometry against the reference QOIs obtained from CFD. The relative error is defined as the percentile difference between the QOIs from the lower-fidelity source and the higher-fidelity source. For each figure, the relative error is styled by the underlying geometry, such that data series with circle, square, and diamond markers indicate the sweep angle of 55°, 65°, and 75°, respectively. The color of the data series, from blue, purple, green, orange, to red, indicates root airfoil shapes of NACA 0010, NACA 0016, NACA 0024, NACA 2416, and NACA 4416, respectively. The solid black line corresponds to the average relative error across all geometries. Across AOAs, the average error increases, with the mean average error being 20.9% and 24.1% for C_L and C_M. The error presented in Fig. 6 serves as the benchmark for LF error. Fig. 7 presents the QOIs relative errors obtained from `VortexNet`, with an identical plotting framework to Fig. 6.

When comparing the QOIs predictions errors from `VortexNet` (Fig. 7) against those from the VLM (Fig. 6), the surrogate model demonstrates strong agreement with the CFD references and provides significantly better predictions than those from the VLM. The mean errors are significantly lower than those reported by the VLM, being −1.3% and 0.3% for C_L and C_M, respectively. Across AOAs, the average

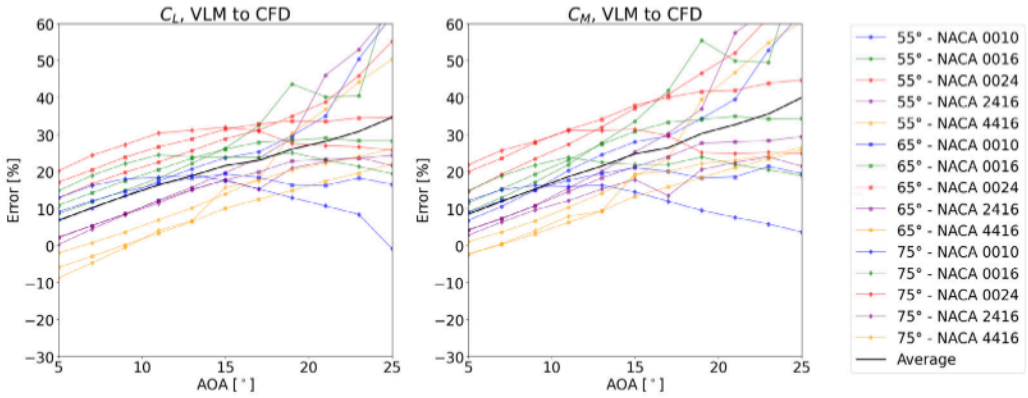


Fig. 6. Relative errors for C_L and C_M from the VLM against the CFD reference.

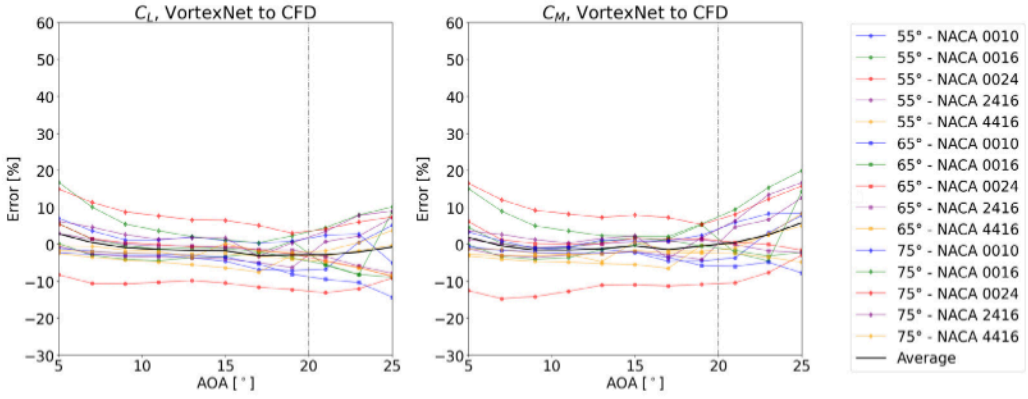


Fig. 7. Relative errors for C_L and C_M from `VortexNet` against the CFD reference. The region to the right of the dash-dotted line (20°) denotes extrapolation beyond the training range.

error obtained from `VortexNet` shows smaller variation and remains relatively close to zero: the average C_L error ranges from −3.1% to 2.8%, and the C_M error ranges from −1.6% to 5.8%. In contrast, the QOIs obtained from VLM show strong dependence on AOA, with the largest average errors observed at 25°, reaching 34.7% and 40.0% for C_L and C_M, respectively. In addition, for each individual geometry, the errors obtained from `VortexNet` exhibit a much weaker dependency on AOA compared to the VLM results, and the inter-geometry error spread is also reduced. Such a characteristic is desired for conceptual design applications, as these error bounds can be incorporated into the design framework. Within the extrapolated region (to the right of the dash-dotted line at 20° in Fig. 7), the average errors for both QOIs increase. However, the deviation remains largely monotonic, and both the error magnitude and the spread across geometries are smaller than those obtained from the VLM (Fig. 6). These improvements, evident in both the trend-wise capturing and the alignment of absolute values for the QOIs, significantly enhance the fidelity of aerodynamic analyses for these platforms.

As the relative error describes the prediction discrepancy for individual samples, using it for this comparison across all free-stream conditions will result in a cumbersome error representation. Instead, we rely on the normalized root-mean-square error (NRMSE) to describe the error, defined as:

$$\mathrm{NRMSE}(f^{\mathrm{method}}) = \frac{1}{\overline{f^{\mathrm{HF}}}} \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i^{\mathrm{method}} - f_i^{\mathrm{HF}})^2}, \tag{24}$$

where f_i^{method} is the QOI evaluated by a specific method, f_i^{HF} is the reference HF QOI obtained from CFD, $\overline{f^{\mathrm{HF}}}$ is the mean of the reference QOI values, and N is the total number of samples in the set. The NRMSE thus represents the deviation of the predicted QOI from the reference value, normalized by the average reference value, allowing for comparison across QOIs.

To highlight the fidelity gain of adopting `VortexNet` in the multi-fidelity workflow, we computed the NRMSEs for QOIs computed using the projected CFD, the `VortexNet` surrogate model predictions, and the VLM, all compared against the CFD reference values. The NRMSEs are grouped by geometries and workflow methods so that, for each wing geometry under a given method, the NRMSE represents an average deviation of predictions across all AOAs, including the small extrapolation region, against CFD references. The results are shown in Fig. 8, with C_L and C_M listed from left to right. The VLM-only errors are displayed in the top row, the `VortexNet` surrogate-augmented errors in the middle row, and the projected CFD errors are shown in the bottom row.

On average, the epistemic discretization error induced by calculating QOIs using projected surface pressures is 3.8%. An examination of the ΔC_p^{HF} field reveals that the majority of this discretization error originates near the wing apex at the central body. This region exhibits large errors due to the use of span-wise cosine spacing, which results in larger panel sizes and excessive averaging of pressure fields with high gradients. The average NRMSEs for C_L and C_M are 4.6% and 3.0%, respectively.

The QOIs computed using the surrogate model have an average NRMSE of 6.1%. Considering that the minimum QOI prediction error using the surrogate model is the epistemic discretization error discussed earlier (since the projected CFD ΔC_p^{HF} is considered as "ground truth" for the surrogate model during training), the fact that this test returns an error close to the epistemic discretization error indicates that the surrogate's modeling error is low. The corresponding NRMSEs for C_L

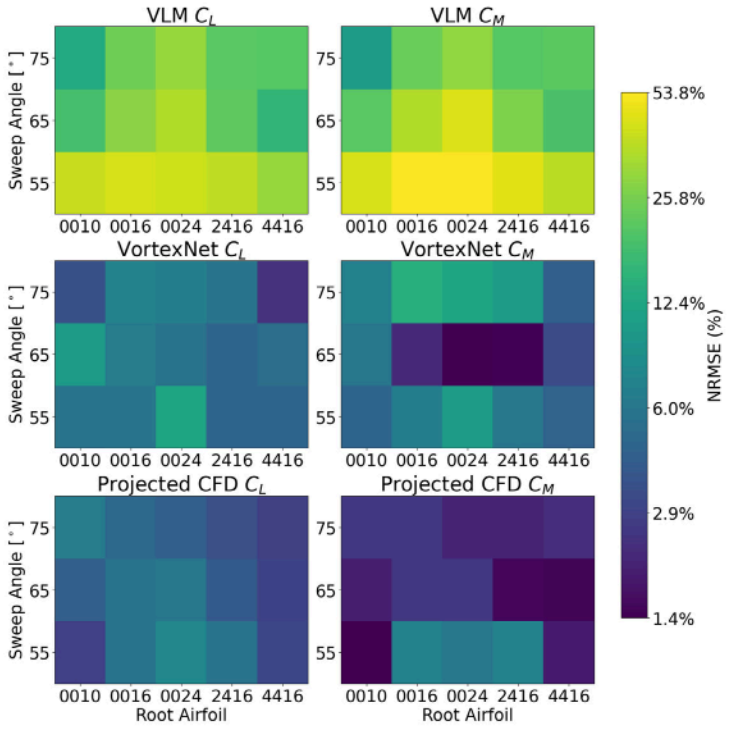


Fig. 8. NRMSEs for C_L and C_M (from left to right, column-wise), comparing QOIs computed by VLM, integrated QOIs from the surrogate model (ΔC_p^{pred}), and QOIs computed from ΔC_p^{HF}, all against the CFD reference. The results are grouped by geometries, with NRMSE values represented by color.

and C_M are 6.0% and 6.3%, respectively. In comparison, the VLM method has an average NRMSE of 30.2%. Specifically, the NRMSEs for C_L and C_M are 28.1% and 32.3%, respectively. Although the database in this test contains extrapolation samples for the surrogate model, the average NRMSE of the QOIs from the proposed mixed-fidelity workflow is still smaller than that of VLM by almost fivefold.

4.2.2. Aerodynamic coefficients prediction for unseen geometries

Our final test case assesses the accuracy of the proposed workflow in predicting QOIs for unseen geometries. By comparing the QOI accuracy against the results presented in the previous section, the model's geometric generalizability can be assessed. We select the four new Delta wing geometries presented in Section 4.1.2. Specifically, two sweep angles, 60° and 70°, and two root airfoils, NACA 0013 and NACA 3416, are used. The free-stream conditions match those reported in Section 4.2.1.

Similar to the relative error comparison in Section 4.2.1, the relative error for the VLM QOIs and `VortexNet` QOIs against the CFD reference is plotted in Figs. 9 and 10. For each figure, the relative error is styled by the underlying geometry, such that data series with circle and square markers indicate the sweep angle being 60° and 70°, respectively. The color of the data series, from blue to red, indicates the root airfoil shape, ranging from NACA 0013 to NACA 3416.

For all unseen geometries, we observe prediction quality and trend-wise characteristics similar to those in Section 4.2.1. The average relative errors from the VLM are 16.0% for C_L and 18.4% for C_M, while `VortexNet` reduces these to −2.0% and −0.7%, respectively. Beyond improving mean accuracy, `VortexNet` also decreases the dispersion of QOIs across geometries and reduces the dependency of QOI error on AOA. Overall, fidelity improvement over the VLM results is observed.

Finally, we compare the NRMSEs for QOIs obtained from the VLM, the `VortexNet` surrogate model, and projected CFD against the CFD references. The results are presented in Fig. 11. On average, the epistemic discretization error induced by calculating QOIs using projected surface pressures is 2.9%, which does not deviate much from the error quantification in Section 4.2.1. The average NRMSEs for C_L and C_M are 4.1% and 1.7%, respectively. Although this error is lower than the corresponding average error in Section 4.2.1, the reduction is most likely due to the fact that the current test involves a database that contains fewer geometric variations.

When `VortexNet` is used for QOI evaluation, we observe an average NRMSE of 3.4%. The NRMSEs for C_L and C_M are 3.5%, and 3.4%, respectively. The difference between the surrogate model-predicted NRMSEs and the projected CFD NRMSEs is comparable to the results of the previous test in Section 4.2.1. We conclude that the workflow does not present significant deviations in QOI prediction accuracy when applied to unseen geometries. For comparison, the average NRMSE for QOIs predicted by the VLM is 24.0%, with NRMSEs for C_L and C_M being 22.4% and 25.7%, respectively. When comparing the average NRMSE obtained from `VortexNet` with that computed by the VLM, a sevenfold reduction in error is observed.

Across all assessments, the proposed surrogate model outperforms the VLM-based approach in QOI prediction accuracy. Additionally, a significant performance degradation is not observed for this dataset after introducing new geometries. This observation supports the proposed workflow's generalizability potential in accommodating geometry variation within the design space. However, a more detailed sensitivity study with a richer design space inclusion is needed in the future to demonstrate the proposed workflow's effectiveness for more complex vehicle conceptual design.

4.3. Prediction explanations

While the proposed `VortexNet` demonstrates strong prediction accuracy and generalizability, the underlying mechanism by which the GNN leverages and aggregates input graph features remains unclear. The complexity of GNNs makes extracting human-intelligible explanations from trained networks challenging [69]. Nevertheless, the authors have conducted a preliminary analysis of the `VortexNet`'s latent space. This section aims to understand how input graph features are utilized across latent layers and provide guidance for refining future `VortexNet`-like GNN architectures.

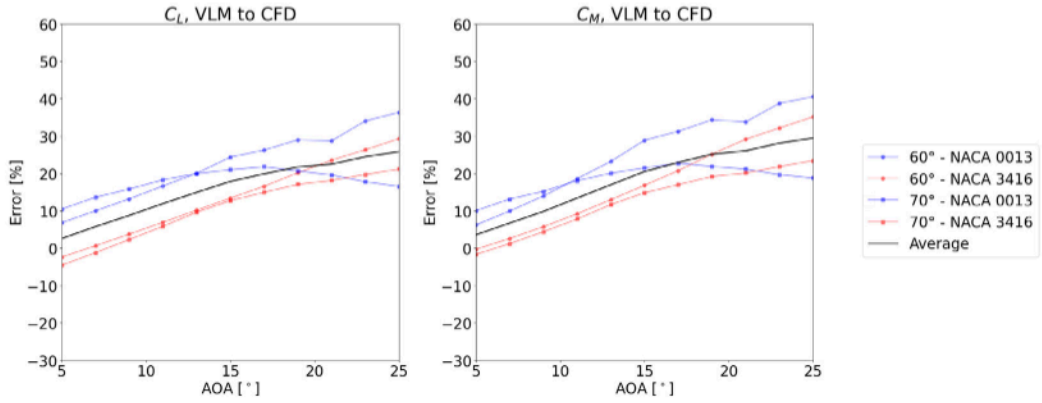


Fig. 9. Relative errors for C_L and C_M from the VLM against the CFD reference for four unseen geometries.

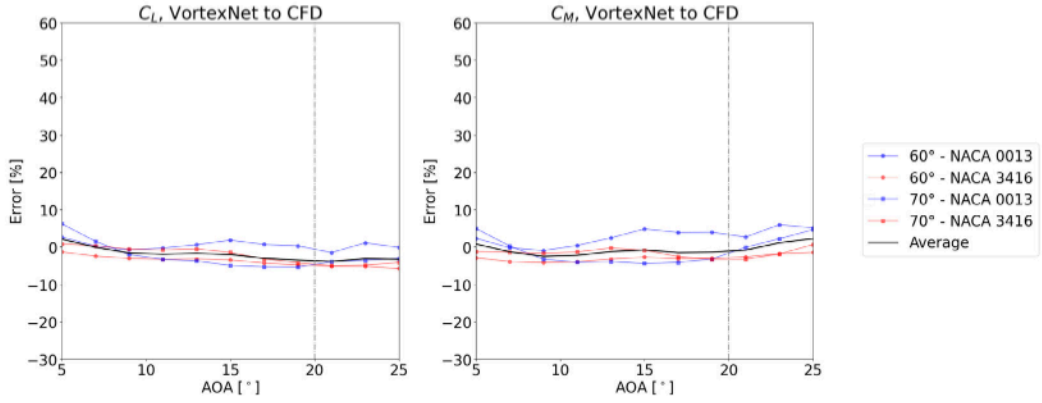


Fig. 10. Relative errors for C_L and C_M from `VortexNet` against the CFD reference for four unseen geometries. The region to the right of the dash-dotted line (20°) denotes extrapolation beyond the training range.

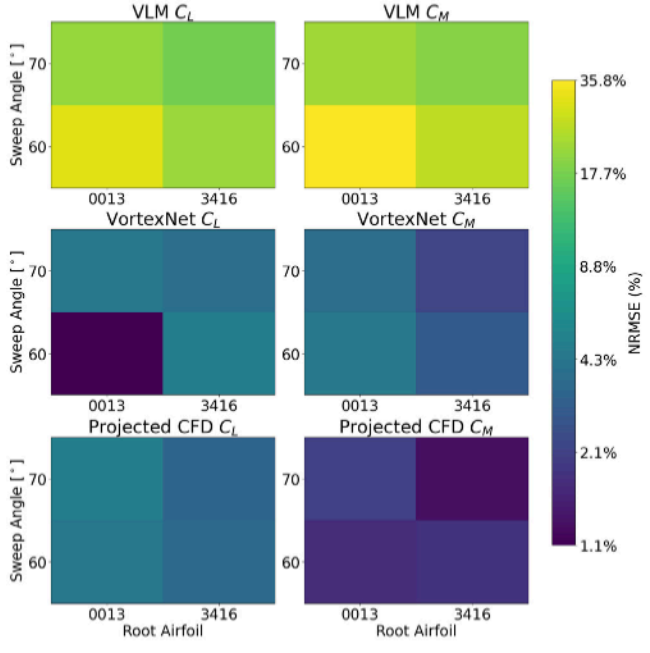


Fig. 11. NRMSEs for C_L and C_M (from left to right, column-wise), comparing QOIs computed by the VLM, integrated QOIs from the surrogate model (ΔC_p^{pred}), and QOIs computed from ΔC_p^{HF}, all against the CFD reference.

Among methods for latent space analysis, we performed an ablation study on latent space features to investigate the resultant ΔC_p predictions. For the data sample, we selected a representative test case featuring a 55° NACA 4416 wing at an AOA of 10.9° and a Mach number of 0.49, as this combination exhibits richer nonlinear fluid dynamics and is therefore well suited for latent space analysis. The study begins with the nodal feature latent space obtained from the output of the first Convolutional Block shown in Fig. 2. At this stage, the nodal feature array was recorded, and clustering was performed on the feature array. Since the node features exist in a high-dimensional space ($\mathbb{R}^{\texttt{Fxheads}}$), the t-SNE dimensionality reduction technique was applied to project the feature space to three dimensions [70]. K-means clustering was then used to classify nodes into four clusters [71], and their corresponding nodal indices were recorded. From the clustering results, we observe that different clusters correspond to distinct regions of the wing. Nodes belonging to "Cluster 1" and "Cluster 2" correspond to the right and left sections of the wing afterbody. Nodes within "Cluster 3" correspond to the central forebody, and nodes in "Cluster 4" correspond to the leading-edge region of the wing. This clusters-to-wing-region association is presented in Fig. 12.

Using the recorded node-to-cluster assignments, the ablation study proceeds by recursively isolating the contribution of each cluster. At a given layer, nodal feature vectors associated with all other clusters are set to zero, thereby retaining only the features of the cluster under investigation. This procedure is repeated across all Convolutional Block layers up to the final Output Decoder layer.

For example, when analyzing the contribution of Cluster 1 after the fourth Convolutional Block (Layer 5), we first extract the nodal feature tensor from the output of Layer 5. The feature vectors corresponding to nodes in all other clusters are then manually zeroed out. The modified

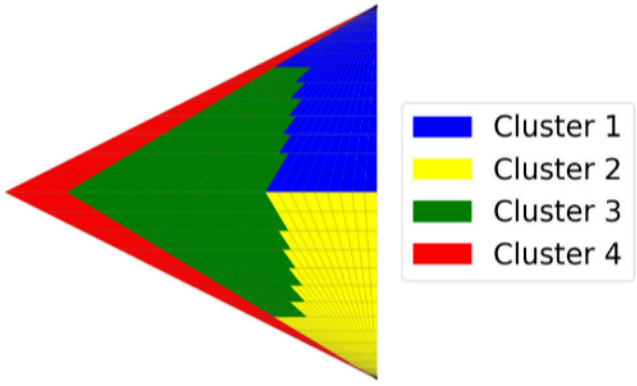


Fig. 12. Association between nodal feature clusters extracted from the first Convolutional Block and their corresponding spatial locations on the wing surface.

global nodal feature tensor is subsequently propagated through the remaining Convolutional Blocks to produce a final modified prediction, denoted as ΔC_p^{mod}. This procedure enables the isolation of the contribution of specific wing-region groups to the overall ΔC_p prediction. By examining the cluster-specific predictions across multiple layers, the progressive effects of multi-hop graph convolution can be analyzed.

The results of the ablation study are presented in Fig. 13. The figure consists of two parts. In the left panel, the ablation errors corresponding to each of the four clusters identified in Fig. 12 are shown in separate columns. The ablation error is defined as the difference between ΔC_p^{mod} and the original `VortexNet` prediction ΔC_p^{pred} computed using the unmodified nodal feature tensor. This ablation error quantifies the influence of a given cluster, ablated at a specific layer, on the final model output.

While the ablation error across all layers is studied, we present selected representative layers rather than all intermediate layers, which would introduce excessive figures without additional insight. Specifically, the ΔC_p^{mod} results for ablations performed at Layer 1 (after the first Convolutional Block), Layer 5 (the intermediate layer, analogous to a bottleneck in an encoder–decoder architecture), and Layer 9 (the final feature prior to the Output Decoder) are plotted. These layers represent key stages in the feature transformation process.

In the right panel, the relationships among nodal features across clusters are visualized. Principal Component Analysis (PCA) is used to project the corresponding nodal features across graphs onto a two-dimensional space visualized by the first two principal components. The scatter points corresponding to each cluster are colored consistently with the color scheme used in Fig. 12.

From the ablation error results, it can be observed that learned field features become increasingly localized across network depth. At shallow layers, removing nodal features from a specific region introduces global effects that extend beyond the corresponding spatial clusters. This behavior is evident in the right and left wing afterbody regions when examining ΔC_p^{mod} at Layer 1 for Cluster 1 and Cluster 2. The removal of surrounding nodal features produces substantial errors, shown as significant negative pressure deviations in regions near the leading edge that extend inside the original regions where nodal features are kept. A similar phenomenon is observed for the ablation error associated with Cluster 3. Interestingly, the ablation error obtained when preserving only Cluster 4 (the leading-edge region) is smaller than the errors corresponding to the other clusters. This suggests that, at shallow layers, the leading-edge region contains more globally informative features that `VortexNet` leverages for pressure field reconstruction. This conjecture is further supported by the distribution of ΔC_p^{LF}, where the highest pressure magnitudes and sharpest gradients are concentrated near the leading edge.

At the intermediate layer (Layer 5), removing specific wing regions no longer induces substantial global error variations. The ablation errors associated with the left wing afterbody from Cluster 1's ΔC_p^{mod}, the right wing afterbody from Cluster 2's, and the central afterbody from Cluster 4's show comparable magnitudes and do not differ significantly. This suggests that a coherent global pressure field representation has already been established by this stage of the network, and truncating features from individual regions primarily produces localized effects. Such localization is further reflected in the sharp spatial boundaries of the ablation errors for each cluster. For Cluster 1 and Cluster 2, we no longer observe the cross-region error propagation seen in earlier layers; eliminating nodes outside a region does not strongly affect predictions within that region. This trend is particularly apparent for Cluster 3, where the ablation error exhibits a clear cutoff along two outboard sections, resembling the geometric extent of the cluster itself, as shown in Fig. 12. This localization effect becomes even more pronounced in deeper layers. At Layer 9, truncating features from a specific region affects only that eliminated region and produces negligible influence on the preserved regions. The preserved regions exhibit near-zero ablation error, indicating that ΔC_p^{mod} within those regions is effectively identical to the original prediction ΔC_p^{pred} obtained without truncation.

This progressive localization of ablation error across layers is consistent with the message-passing mechanism of GNNs, in which nodal features are iteratively aggregated from neighboring nodes at each hop. For a preserved region to achieve accurate predictions, it must incorporate information from adjacent regions, especially from the leading edge region in the present study. Once this information propagates into the preserved region, it is further aggregated locally to refine the field prediction. Because `VortexNet` is built upon a GNN architecture, information propagation across the graph is governed by the number of message-passing hops, implemented as stacked Convolutional Block layers. For larger graphs with finer panel discretizations, deeper networks would therefore be required to ensure that information from physically informative regions can traverse sufficiently across the graph and influence distant nodes.

However, deeper networks are more challenging to train and prone to high-variance error. To address this limitation, improved edge definitions that connect nodes beyond direct neighbors could be implemented. Alternatively, using a fully connected graph may help mitigate this issue, though the implications of adopting such a graph remain unclear as the addition of trainable parameters may decrease the prediction accuracy under limited HF samples. Additionally, since pooling layers are not utilized in this network, we believe that incorporating a properly designed pooling and unpooling mechanism, such as the one proposed by Immordino et al. [59], could assist information propagation by reducing the computational graph size toward deeper layers. This highlights the need for future research on optimizing graph definitions and model architectures in `VortexNet`-like surrogate modeling techniques.

Finally, we examine the feature representations shown in the right panel of Fig. 13. In the shallower layers, the distribution of nodal feature clusters loosely follows the positional distribution of the corresponding nodes in physical space. For example, the relative separation between the left and right afterbody sections can still be identified along the principal axes. As depth increases, this grouping by physical proximity diminishes. By Layer 5, the majority of nodes collapse into a single dominant cluster, with only a subset of leading-edge nodes remaining separated. This behavior suggests that, at this stage, features associated with flow physics become more influential than purely geometric location. Because the left and right outboard sections experience symmetric free-stream conditions, their physical separation becomes a less dominant organizing factor in clustering. Additionally, from Layer 1 to Layer 5, the overall aspect ratio and spatial spread of the nodal feature distribution decrease, as evidenced by the reduced extent of the combined cluster manifold. This contraction of the latent

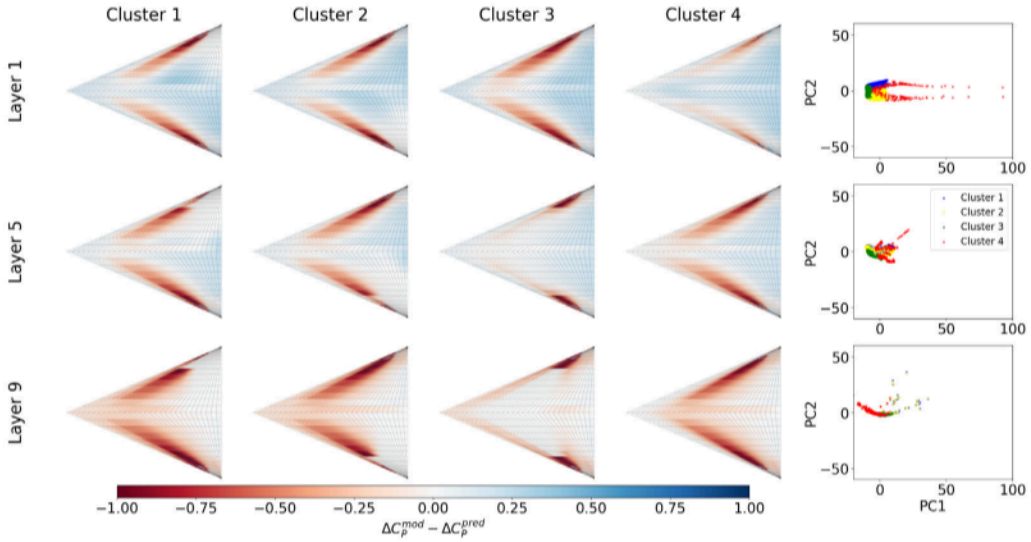


Fig. 13. Ablation study results illustrating ablation error ($\Delta C_p^{\text{mod}} - \Delta C_p^{\text{pred}}$) for node features within each cluster (columns), and the projected latent space nodal features plotted against the principal axes, with scatter dot colors indicating the corresponding clusters of the nodes. The rows represent the specific layers where the ablation study is conducted.

space suggests a potential mechanism underlying `VortexNet`'s generalizability: by nonlinearly mapping the input geometry–flow state into a more compact latent representation, the network facilitates smoother interpolation across nodes. Such compactness may enable consistent field prediction across varying geometries and flow conditions. At the last layer, the clusters re-expand while preserving the left–right symmetry between Cluster 1 and Cluster 2, further supporting the conjectured mechanism between geometric structure and flow-conditioned feature organization in the learned latent space.

5. Conclusions and future work

In this work, we propose a novel GNN-based surrogate model named `MF-VortexNet` for aircraft conceptual design. The surrogate model integrates high-fidelity CFD data into field data predictions generated by low-fidelity prediction tools, such as the vortex lattice method. Compared to traditional surrogate modeling techniques, which primarily focus on specific scalar variables, the field data prediction capability allows for arbitrary quantities of interest to be evaluated within conceptual design environments. Once trained, the surrogate model can generate quasi-HF field data predictions that capture the pressure field impacted by nonlinear flow physics at a cost comparable to that of running a VLM, thereby addressing the fidelity-cost trade-off challenge in the conceptual design phase.

The `MF-VortexNet` architecture utilizes U-net-style skip connections for global feature capturing and employs graph attention network (GAT) Convolutional Blocks for local flow feature capture, such as vortex lift and flow separation. Data assembly methods, implicit physical constraint enforcement, and training schemes tailored to `MF-VortexNet` are also explored and presented.

The model's prediction accuracy is evaluated both qualitatively and quantitatively. When provided with an input geometry seen during training, the surrogate model demonstrates strong performance in reconstructing qualitative flow features and achieves low prediction RMSE, establishing significant fidelity improvements over LF-based aerodynamic prediction methods. Deployment of `MF-VortexNet` for lift and moment coefficient prediction yields a five- to seven-fold reduction in NRMSE relative to the LF solver, depending on the geometry set examined. The model's generalizability is also evaluated, demonstrating its ability to predict pressure fields and recover integrated aerodynamic quantities for unseen geometries. Nonetheless, a degradation in field reconstruction quality is observed relative to seen geometries, indicating opportunities for future improvement. Despite these limitations, the model demonstrates potential as a mesh-agnostic surrogate. Such capability is critical for conceptual aircraft design under design space exploration. Finally, a latent space ablation study is conducted to explain the prediction mechanism of the proposed network. The results indicate that `MF-VortexNet` attains its predictive capability by jointly encoding geometric and flow features into a compact, interpolatable latent space.

Future work will focus on enhancing the model's generalizability and improving feature-capturing efficiency for more complex geometries, graph representations, and broader aerodynamic conditions. We believe that the model's predictive capability and efficiency can be further improved through the introduction of hierarchical pooling mechanisms, refinement of edge-connectivity graphs, and the use of more representative and diverse training datasets. As the proposed framework is not restricted to pressure-field regression, its application on cross-fidelity bridging can extend beyond pressure field and aerodynamics modeling. A promising direction is the extension of the proposed surrogate framework to predict field quantities that are poorly captured or absent in classical low-fidelity methods, such as wall-shear stress, local boundary-layer properties, and thermal loads. Finally, as a data-driven surrogate modeling framework, `MF-VortexNet` is not limited to the VLM–RANS setting considered in this study and can be generalized to additional fidelity levels. Extending the model to interpolate across a broader range of fidelity levels is a promising direction for future applications.

CRediT authorship contribution statement

Yiren Shen: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization. **Jacob T. Needels:** Writing – review & editing, Writing – original draft, Investigation. **Juan J. Alonso:** Writing – review & editing, Supervision, Project administration, Conceptualization, Investigation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yiren Shen, Jacob Needels, Juan Alonso reports financial support was

provided by Air Force Office of Scientific Research. Yiren Shen, Jacob Needels, Juan Alonso reports financial support was provided by NASA. Yiren Shen reports equipment, drugs, or supplies was provided by Google LLC. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Air Force Office of Scientific Research under Grant FA9550-22-1-0004 and NASA, Award 80NSSC1 9K1661, under the Commercial Supersonics Technology (CST) program, Supersonic Configurations at Low Speeds (SCALOS), with Sarah Langston as the NASA technical grant monitor. The authors extend their gratitude to Eli Livne, Kuang-Ying Ting, Chester Nelson, Kenneth Wiersema, Aurelien Ghiglino, and Emilio Botero for their valuable discussions on supersonic aircraft configurations and conceptual design tools. Additionally, the authors would like to thank the Stanford Research Computing and the Stanford Data Science for providing computational resources on the Sherlock and the Marlowe clusters [72], as well as the Google Cloud Research Credit Program for granting access to GPU resources on the Google Cloud Platform.

References

[1] Brunton SL, Noack BR, Koumoutsakos P. Machine Learning for Fluid Mechanics. Annu Rev Fluid Mech 2020;52:477–508. http://dx.doi.org/10.1146/annurev-fluid-010719-060214.

[2] Shen Y, Patel HC, Xu Z, Alonso JJ. Application of multi-fidelity transfer learning with autoencoders for efficient construction of surrogate models. In: AIAA SCITECH 2024 forum. (January):2024, p. 1–19. http://dx.doi.org/10.2514/6.2024-0013.

[3] Neufeld D, Chung J, Behdinian K. Aircraft conceptual design optimization considering fidelity uncertainties. J Aircr 2011;48(5):1602–12. http://dx.doi.org/10.2514/1.C031312.

[4] Raymer DP. Aircraft design: A conceptual approach, In: AIAA education series, fourth ed.. Reston, VA: American Institute of Aeronautics and Astronautics; 2006.

[5] Martins JRRA, Lambe AB. Multidisciplinary design optimization: A survey of architectures. AIAA J 2013;51(9):2049–75. http://dx.doi.org/10.2514/1.J051895.

[6] Wang L, Diskin B, Biedron RT, Nielsen EJ, Bauchau OA. High-fidelity multidisciplinary sensitivity analysis and design optimization for rotorcraft applications. AIAA J 2019;57(8):3117–31. http://dx.doi.org/10.2514/1.J056587.

[7] Wendorff A, Variyar A, Ilario C, Botero E, Capristan F, Smart J, Alonso J, Kulik L, Clarke M, Colonno M, Kruger M, Vegh JM, Goncalves P, Erhard R, Fenrich R, Orra T, St. Francis T, MacDonald T, Momose T, Economon T, Lukaczyk T, Maier W. SUAVE: An aerospace vehicle environment for designing future aircraft. 2020, GitHub repository URL https://github.com/suavecode/SUAVE.

[8] MacDonald T, Clarke M, Botero EM, Vegh JM, Alonso JJ. SUAVE: An open-source environment enabling multi-fidelity vehicle optimization. In: 18th AIAA/ISSMO multidisciplinary analysis and optimization conference. 2017, http://dx.doi.org/10.2514/6.2017-4437.

[9] Owens D. Weissinger's model of the nonlinear lifting-line method for aircraft design. In: 36th AIAA aerospace sciences meeting and exhibit. 1998, http://dx.doi.org/10.2514/6.1998-597.

[10] Budziak K. Aerodynamic Analysis with Athena Vortex Lattice (AVL). Hamburg: Aircraft Design and Systems Group (AERO), Department of Automotive and Aeronautical Engineering, Hamburg University of Applied Sciences; 2015, URL https://www.fzt.haw-hamburg.de/pers/Scholz/arbeiten/TextBudziak.pdf.

[11] Miranda LR, Elliott RD, Baker WM. A generalized vortex lattice method for subsonic and supersonic flow applications. NASA. Langley Res Cent Vortex-Lattice Util 1977. URL https://ntrs.nasa.gov/citations/19780008059.

[12] Charles SW, Ishwar BC. Application of the vortex-lattice technique to the analysis of thin wings with vortex separation and thick multi-element wings. In: Vortex-lattice utilization, a workshop held at Langley Research Center. (May). 1976, URL https://ntrs.nasa.gov/api/citations/19760021075/downloads/19760021075.pdf.

[13] Joshi H, Thomas P. Review of vortex lattice method for supersonic aircraft design. Aeronaut J 2023;1–35. http://dx.doi.org/10.1017/aer.2023.25.

[14] Han Z, Qiao J, Zhang L, Chen Q, Yang H, Ding Y, Zhang K, Song W, Song B. Recent progress of efficient low-boom design and optimization methods. Prog Aerosp Sci 2024;146:101007. http://dx.doi.org/10.1016/j.paerosci.2024.101007.

[15] Seraj S, Martins JR. Aerodynamic shape optimization of a supersonic transport considering low-speed stability. In: AIAA SCITECH 2022 forum. 2022, http://dx.doi.org/10.2514/6.2022-2177.

[16] Nelson CP, Ting KY, Mavriplis N, Soltani RM, Livne E. Supersonic Configurations at Low Speeds (SCALOS): Project Background and Progress at University of Washington. AIAA SCITECH 2022 Forum 2022. http://dx.doi.org/10.2514/6.2022-1803.

[17] Ting K-Y, Nelson C, Wiersema K, Livne E. Supersonic configurations at low speeds (SCALOS): Wind tunnel progress at university of washington. In: AIAA SCITECH 2025 forum. 2025, http://dx.doi.org/10.2514/6.2025-2359.

[18] Polhamus EC. A concept of the vortex lift of sharp-edge delta wings based on a leading-edge-suction analogy. NASA. Langley Res. Center; 1966, URL https://ntrs.nasa.gov/citations/19670003842.

[19] Huynh D, Di Pasquale D, Prince S, Ahuja V. Application of a semi-empirical method to model subsonic vortex lift over sharp leading-Edge Delta wings. In: AIAA SCITECH 2023 forum. 2023, http://dx.doi.org/10.2514/6.2023-2457.

[20] Miller DS, Wood RM. Leeside flows over delta wings at supersonic speeds. J Aircr 1984;21(9):680–6. http://dx.doi.org/10.2514/3.45014.

[21] Mitchell AM, Morton SA, Forsythe JR, Cummings RM. Analysis of delta-wing vortical substructures using detached-eddy simulation. AIAA J 2006;44(5):964–72. http://dx.doi.org/10.2514/1.755.

[22] Di Fabbio T, Rajkumar K, Tangermann E, Klein M. Towards the understanding of vortex breakdown for improved rans turbulence modeling. Aerosp Sci Technol 2024;146:108973. http://dx.doi.org/10.1016/j.ast.2024.108973.

[23] Dacles-Mariani J, Zilliac GG, Chow JS, Bradshaw P. Numerical/experimental study of a wingtip vortex in the near field. AIAA J 1995;33(9):1561–8. http://dx.doi.org/10.2514/3.12826.

[24] Widhalm M, Stradtner M, Schütte A, Ghoreyshi M, Jirásek A, Seidel J. Stability derivatives prediction using reduced-order models on Triple Delta wing. J Aircr 2025;1–14. http://dx.doi.org/10.2514/1.C038321.

[25] Kennedy MC, O'Hagan A. Bayesian calibration of computer models. J R Stat Soc Ser B Stat Methodol 2001;63(3):425–64. http://dx.doi.org/10.1111/1467-9868.00294.

[26] Mukhopadhaya J, Whitehead BT, Quindlen JF, Alonso JJ, Cary AW. Multi-fidelity modeling of probabilistic aerodynamic databases for use in aerospace engineering. Int J Uncertain Quantif 2020;10(5):425–47. http://dx.doi.org/10.1615/Int.J.UncertaintyQuantification.2020032841.

[27] Berkooz G, Holmes P, Lumley JL. The proper orthogonal decomposition in the analysis of turbulent flows. Annu Rev Fluid Mech 1993;25:539–75. http://dx.doi.org/10.1146/ANNUREV.FL.25.010193.002543.

[28] Mrosek M, Othmer C, Radespiel R. Reduced-order modeling of vehicle aerodynamics via proper orthogonal decomposition. SAE Int J Passeng Cars - Mech Syst 2019;12:225–36. http://dx.doi.org/10.4271/06-12-03-0016.

[29] Chang C-M, Tsai H-Y, Chiu T-Y, Cheng C-J, Lin W-A, Chen R-B, Chou Y-J. Proper-orthogonal-decomposition–based emulation of spatiotemporal evolution of turbulent wakes. AIAA J 2025;63(9):3582–94. http://dx.doi.org/10.2514/1.J064779.

[30] Swischuk R, Mainini L, Peherstorfer B, Willcox K. Projection-based model reduction: Formulations for physics-based machine learning. Comput Fluids 2019;179:704–17. http://dx.doi.org/10.1016/j.compfluid.2018.07.021.

[31] Perron C, Rajaram D, Mavris D. Development of a multi-fidelity reduced-order model based on manifold alignment. In: AIAA AVIATION 2020 FORUM. http://dx.doi.org/10.2514/6.2020-3124.

[32] Schwerdtner P, Peherstorfer B. Greedy construction of quadratic manifolds for nonlinear dimensionality reduction and nonlinear model reduction. In: arXiv preprint. 2024, http://dx.doi.org/10.48550/arXiv.2403.06732.

[33] Decker K, Schwartz HD, Mavris D. Dimensionality reduction techniques applied to the design of hypersonic aerial systems. In: AIAA AVIATION 2020 FORUM. http://dx.doi.org/10.2514/6.2020-3003.

[34] Decker K, Iyengar N, Rajaram D, Perron C, Mavris D. Manifold alignment-based nonintrusive and nonlinear multifidelity reduced-order modeling. AIAA J 2023;61(1):454–74. http://dx.doi.org/10.2514/1.J061720.

[35] Black N, Najafi AR. Learning finite element convergence with the Multi-fidelity Graph Neural Network. Comput Methods Appl Mech Engrg 2022;397:115120. http://dx.doi.org/10.1016/j.cma.2022.115120.

[36] Li J, Li Y, Liu T, Zhang D, Xie Y. Multi-fidelity graph neural network for flow field data fusion of turbomachinery. Energy 2023;285(July):129405. http://dx.doi.org/10.1016/j.energy.2023.129405.

[37] Chen J, Hachem E, Viquerat J. Graph neural networks for laminar flow prediction around random two-dimensional shapes. Phys Fluids 2021;33(12). http://dx.doi.org/10.1063/5.0064108.

[38] Pfaff T, Fortunato M, Sanchez-Gonzalez A, Battaglia P. Learning mesh-based simulation with graph networks. In: ICLR. 2021, https://openreview.net/forum?id=roNqYL0_XP.

[39] Shao X, Liu Z, Zhang S, Zhao Z, Hu C. PIGNN-CFD: A physics-informed graph neural network for rapid predicting urban wind field defined on unstructured mesh. Build Environ 2023;232:110056. http://dx.doi.org/10.1016/j.buildenv.2023.110056.

[40] Yang S, Vinuesa R, Kang N. Enhancing graph U-nets for mesh-agnostic spatio-temporal flow prediction. In: arXiv preprint. 2024, arXiv:2406.03789.

[41] Liu Q, Zhu W, Jia X, Ma F, Gao Y. Fluid simulation system based on graph neural network. In: arXiv preprint. 2022, arXiv:2202.12619.

[42] Gao H, Ji S. Graph U-nets. In: Chaudhuri K, Salakhutdinov R, editors. Proceedings of the 36th international conference on machine learning, ICML 2019, 9-15 June 2019, long beach, california, USA. Proceedings of machine learning research, vol. 97, PMLR; 2019, p. 2083–92, URL http://proceedings.mlr.press/v97/gao19a.html.

[43] Wei L, Freris NM. Multi-scale graph neural network for physics-informed fluid simulation. Vis Comput 2024. http://dx.doi.org/10.1007/s00371-024-03402-6.

[44] Shen Y, Needels JT, Alonso JJ. VortexNet: A graph neural network-based multi-fidelity surrogate model for field predictions. In: AIAA SCITECH 2025 forum. 2025, http://dx.doi.org/10.2514/6.2025-0494.

[45] Shen Y, Alonso JJ. Performance evaluation of a graph neural network-augmented multi-fidelity workflow for predicting aerodynamic coefficients on delta wings at low speed. In: AIAA SCITECH 2025 forum. 2025, http://dx.doi.org/10.2514/6.2025-2360.

[46] Economon TD, Palacios F, Copeland SR, Lukaczyk TW, Alonso JJ. SU2: An open-source suite for multiphysics simulation and design. AIAA J 2016;54(3):828–46. http://dx.doi.org/10.2514/1.J053813.

[47] Botero EM, Clarke MA, Erhard RM, Smart JT, Alonso JJ, Blaufox A. Aerodynamic verification and validation of SUAVE. In: AIAA SCITECH 2022 forum. 2022, http://dx.doi.org/10.2514/6.2022-1929.

[48] Bertin JJ, Cummings RM. Aerodynamics for engineers. sixth ed.. Cambridge, England: Cambridge University Press; 2021, p. 354–62.

[49] Drela M. Flight Vehicle Aerodynamics. Cambridge, MA: The MIT Press; 2014, p. 127–35.

[50] Palacios F, Alonso J, Duraisamy K, Colonno M, Hicken J, Aranake A, Campos A, Copeland S, Economon T, Lonkar A, et al. Stanford university unstructured (SU2): an open-source integrated computational environment for multi-physics simulation and design. In: 51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. 2013, p. 287. http://dx.doi.org/10.2514/6.2013-287.

[51] Palacios F, Economon TD, Aranake A, Copeland SR, Lonkar AK, Lukaczyk TW, Manosalvas DE, Naik KR, Padron S, Tracey B, et al. Stanford university unstructured (SU2): Analysis and design technology for turbulent flows. In: 52nd aerospace sciences meeting. 2014, p. 0243. http://dx.doi.org/10.2514/6.2014-0243.

[52] Spalart P, Allmaras S. A one-equation turbulence model for aerodynamic flows. In: 30th aerospace sciences meeting and exhibit. 1992, http://dx.doi.org/10.2514/6.1992-439.

[53] Seraj S, Martins JRRA. Predicting the high-angle-of-attack characteristics of a delta wing at low speed. J Aircr 2022;59(4):1071–81. http://dx.doi.org/10.2514/1.C036618.

[54] Jameson A, Schmidt W, Turkel E. Numerical solution of the Euler equations by finite volume methods using runge kutta time stepping schemes. In: 14th fluid and plasma dynamics conference. 1981, http://dx.doi.org/10.2514/6.1981-1259.

[55] McDonald RA, Gloudemans JR. Open vehicle sketch pad: An open source parametric geometry and analysis tool for conceptual aircraft design. In: AIAA SCITECH 2022 forum. 2022, http://dx.doi.org/10.2514/6.2022-0004.

[56] Hamilton WL, Ying R, Leskovec J. Representation learning on graphs: Methods and applications. 2017, CoRR abs/1709.05584 arXiv:1709.05584.

[57] Zhao Y, Li H, Zhou H, Attar HR, Pfaff T, Li N. A review of graph neural network applications in mechanics-related domains. Artif Intell Rev 2024;57(11):315. http://dx.doi.org/10.1007/s10462-024-10931-y.

[58] Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw. Learn Syst 2021;32(1):4–24. http://dx.doi.org/10.1109/tnnls.2020.2978386.

[59] Immordino G, Vaiuso A, Da Ronch A, Righi M. Predicting transonic flowfields in non–homogeneous unstructured grids using autoencoder graph convolutional networks. J Comput Phys 2025;524:113708. http://dx.doi.org/10.1016/j.jcp.2024.113708.

[60] Deshpande S, Bordas SP, Lengiewicz J. MAgNET: A graph U-net architecture for mesh-based simulations. Eng Appl Artif Intell 2024;133:108055. http://dx.doi.org/10.1016/j.engappai.2024.108055.

[61] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph Attention Networks. Int Conf Learn Represent 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

[62] Brody S, Alon U, Yahav E. How attentive are graph attention networks?. In: ICLR. 2022, URL https://arxiv.org/abs/2105.14491 arXiv:2105.14491.

[63] Anguita D, Ghelardoni L, Ghio A, Oneto L, Ridella S, et al. The'k'in K-fold cross validation.. In: ESANN. 102, 2012, p. 441–6, URL https://www.esann.org/sites/default/files/proceedings/legacy/es2012-62.pdf.

[64] PyTorch Contributors. Smoothl1loss. In: PyTorch 2.5 Documentation. 2024, Retrieved 30 November 2024. URL https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html.

[65] Eivazi H, Tahani M, Schlatter P, Vinuesa R. Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. Phys Fluids 2022;34(7):075117. http://dx.doi.org/10.1063/5.0095270.

[66] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Bengio Y, LeCun Y, editors. 3rd international conference on learning representations, ICLR 2015, san diego, CA, USA, May 7-9, 2015, conference track proceedings. 2015, URL http://arxiv.org/abs/1412.6980.

[67] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. KDD '19, New York, NY, USA: Association for Computing Machinery; 2019, p. 2623–31. http://dx.doi.org/10.1145/3292500.3330701.

[68] Pointwise. retrived 26 April 2023. Pointwise, Cadence Design Systems. URL http://www.pointwise.com/.

[69] Ying R, Bourgeois D, You J, Zitnik M, Leskovec J. Gnnexplainer: Generating explanations for graph neural networks. In: arXiv preprint. 2019, arXiv:1903.03894.

[70] Van der Maaten L, Hinton G. Visualizing data using t-sne.. J Mach Learn Res 2008;9(86):2579–605, URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[71] MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proceedings of 5-th berkeley symposium on mathematical statistics and probability/university of california press. 1967, URL https://zbmath.org/?q=an:0214.46201.

[72] Kapfer C, Stine K, Narasimhan B, Mentzel C, Candes E. Marlowe: Stanford's GPU-based computational instrument. In: Zenodo. 2025, http://dx.doi.org/10.5281/zenodo.14751899.