Aerospace Science and Technology 164 (2025) 110414

Contents lists available at ScienceDirect

Aerospace Science and Technology

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

Original article

Accelerating fluid simulations with graph convolution network predicted flow fields

Wei Xian Lim [a], Loh Sher En Jessica [b], Yu Lv [c], Adams Wai-Kin Kong [a,b], Wai Lee Chan [a,d,*]

[a] Rolls-Royce@NTU Corporate Lab, Nanyang Technological University, Singapore, 637460, Singapore
[b] College of Computing and Data Science, Nanyang Technological University, Singapore, 639798, Singapore
[c] State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing, 100190, China
[d] School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, 639798, Singapore

ARTICLE INFO

Communicated by Jiaqing Kou

Keywords:
Deep learning
Graph convolution networks
Reynolds-averaged Navier–Stokes simulations
Flow initialization

ABSTRACT

The field of computational fluid dynamics (CFD) is integral to engineering disciplines, particularly for designing systems that operate under complex fluid flow conditions. Accurate simulation of flow fields is essential for optimizing performance across a variety of applications, including aviation, automotive, marine, and renewable energy sectors. Recent advancements in deep learning, particularly graph convolution networks (GCNs), offer promising alternatives for improving simulation processes. This work introduces a novel approach to accelerating fluid simulations using GCNs for flow field initialization. To this end, two different GCN models were employed, incorporating prior knowledge of the problem like its boundary conditions, as well as residual training. Extensive experiments using over 2000 sets of simulation results of various NACA airfoil shapes and flow conditions demonstrate that GCN-based initialization significantly reduces computational resources while maintaining high accuracy, achieving a 30% – 50% reduction in simulation time compared to conventional CFD initialization method.

1. Introduction

The field of computational fluid dynamics (CFD) is perpetually evolving, driven by the relentless pursuit to accurately model flow fields around complex geometries. This pursuit is not only of academic interests, but is also grounded in the practical need to predict the effects of fluid flows on various engineering applications, such as the aerodynamics performance of aircrafts and automobiles, characterization of propulsors and thrusters, and adequacy of dispersion and ventilation. The complexity of these systems, characterized by intricate geometries, flow separation, reattachment phenomena [1], and multi-physics interactions [2], poses significant challenges [3] to traditional CFD approaches [4,5].

The conventional methodologies in CFD have been rigorously developed over the years. However, they often grapple with resource limitation when it comes to handling the nuanced interactions in complex flow scenarios, such as high-lift devices where convoluted geometries are typically involved. Traditional grid-based numerical methods, as highlighted by Refs. [6,7], normally require high-resolution meshes to accurately capturing complex fluid-surface interactions, which is crucial for realistic flow predictions, yet computationally expensive. While various approximation techniques have been developed to enhance computational efficiency or accelerate the CFD processes, such as coarse meshing, simplified turbulence modeling [8,9], and reduced-order modeling [10], they often come at the expense of solution fidelity. Therefore, a balance between accuracy and computational cost is necessary.

In contrast to traditional methods, deep learning offers a promising avenue to address the inherent challenges in CFD. With the aid of deep learning, speeding up the entire simulation processes can be done from various perspectives such as developing data-driven turbulence modeling [11], utilizing neural networks to build a reduced-order model for unsteady aerodynamic predictions [12,13], and coupling physics solver and neural networks to offer super-resolution capability [14–16]. Specifically, the integration of convolutional neural networks (CNNs) in CFD [15,17–19] has been a groundbreaking advancement in facilitating

* Corresponding author at: School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore, 639798, Singapore.
E-mail address: chan.wl@ntu.edu.sg (W.L. Chan).

https://doi.org/10.1016/j.ast.2025.110414
Received 12 December 2024; Received in revised form 27 May 2025; Accepted 2 June 2025
Available online 9 June 2025

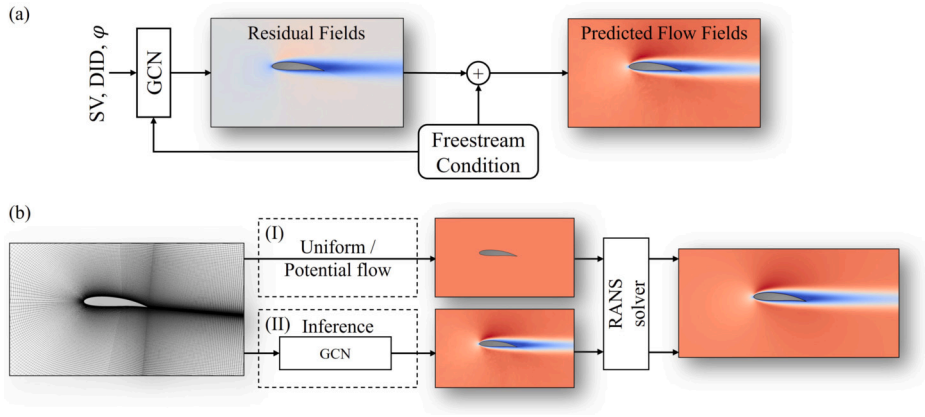

Fig. 1. (a) GCN training pipeline that considers geometrical information (SV and DID) and inlet flow condition (φ) as network inputs and utilizes residual training scheme with freestream condition as the estimator. (b)(I) Typical initialization with uniform or potential flow fields; and (II) Initialization with predicted flow fields by the trained GCN model; both using RANS solver to converge towards steady-state solutions. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

fluid simulations due to the ability of CNNs to process grid-like data, which aligns well with the structured mesh used in CFD simulations.

Among the latest developments, graph convolutional networks (GCNs) have emerged as a powerful tool for flow field prediction [20–28]. Their inherent ability to represent data as graphs, where nodes denote entities and edges describe relationships, renders them compatible with the finite-volume method and hence well-suited for capturing complex flow fields commonly encountered in CFD applications. References [21,22,28] have been effective in handling single geometry in fluid dynamics by leveraging the flexibility of graph-based representations. However, many of them directly predict the flow variables, such as velocity and pressure, and mostly serve as an end-to-end surrogate [20,26]. One exception is the work of Obiols-Sales et al. [17] where fields were treated as images, which does not guarantee meeting the critical convergence criteria of CFD simulations. In addition, only subsets of the flow fields were predicted in most of the current works [24], leading to a lack of comprehensive flow information.

The current challenge lies in predicting physically plausible flow fields from a given geometry that can be used to initialize CFD simulations and reduce computational cost. Unlike previous studies [17,20,24,26] that either rely on surrogate models trained to replace CFD solvers entirely, or treat fields as images, or consider partial domains, or generate approximate fields without direct integration into numerical solvers, the present study focuses on using GCN-based models to produce physically consistent flow field predictions for initialization. Although the idea of using neural network-predicted flow fields to accelerate CFD simulations has been explored in prior works, the methodology adopted here offers several important distinctions. Specifically, the learning task was formulated as a residual training technique [29], where the models estimate deviations from freestream conditions instead of directly predicting absolute values, enhancing stability across diverse flow scenarios. In addition, the geometric representations of embedded solid bodies were implemented through shortest vector and directional integrated distance [30], enabling generalization across a wide range of airfoil geometries. The approach was also validated across two structurally different GCN models, namely MeshGraphNet (MGN) and Invariant Edge-GCN (IVE), demonstrating its architecture-agnostic applicability. Most importantly, the predicted fields are solved with a conventional RANS solver, and their impact is quantitatively assessed based on convergence behavior and computational time reduction. Through these features, the proposed method was able to reduce the computational time required to achieve steady-state solutions in Reynolds averaged Navier–Stokes simulations of flow over NACA airfoil while maintaining high prediction accuracy.

The remainder of this paper consists of four sections. In Sec. 2, the network architecture with its various features are described, followed by the discussion on the generation of datasets for training, validating, and testing purposes in Sec. 3. The experiment results are discussed in Sec. 4. Lastly, conclusion and future work are presented in Sec. 5.

2. Neural network architecture

Graph convolution network is utilized as the fundamental network due to its graph structure characteristics, which enable the treatment of cell centers and faces as nodes and edges, respectively. This allows the neural network to efficiently interpret and process geometric data. For instance, the cell centers and faces in a computational mesh are the counterparts of the nodes, V, and bi-directional edges, E, which describe the connectivity between the graph nodes. Two GCNs, MeshGraphNet (MGN) [21] and Invariant Edge-GCN (IVE) [22] were adopted to demonstrate this deep learning approach of CFD field initialization is general to GCN architectures.

MGN is a message-passing GCN that has demonstrated strong performance in simulating physical systems involving mesh-based data [21]. IVE, on the other hand, integrates geometric invariance and edge-level message passing, which improves prediction accuracy around complex geometries [22]. These two models were chosen due to their complementary characteristics: MGN offers architectural simplicity and scalability, while IVE introduces invariant design principles well-suited for CFD problems with diverse geometrical variations. Their inclusion allows one to assess the generality of the proposed GCN-based flow field initialization approach across different graph-based neural architectures.

Fig. 1(a) illustrates a neural network training pipeline for fluid dynamics using GCNs. The training scheme consists of two components: (i) Network inputs, which are inlet flow conditions ($\varphi = \{u, v, p\}$) and geometry representation of the embedded body (in this case, an airfoil) via shortest Vector (SV) and directional integrated distance (DID) [30]; and (ii) Residual training technique with freestream conditions as the estimator. The SV and DID methods were utilized to encode geometries as input node features. In brief, SV provides both distance and direction information by a vector formed between a node and the nearest surface node of the body. Likewise, the DID defines the perception of the body relative to a node averaged over a prescribed angular range. With the network inputs, the residual flow fields are then predicted via the residual training technique and combined with the freestream condition, thus returning a prediction of the flow fields around the embedded body. The details of the residual training technique and the neural network inputs and parameters are given in subsequent sections.

Traditionally, the initial conditions are described by uniform or potential flow around the airfoils, as illustrated in Fig. 1(b)(I). Engaging RANS solver ensures the final steady-state solutions sufficiently satisfy

Table 1
List of neural network parameters used.

	MGN	IVE
Number of hidden layers	15	8
Hidden layer size (node)	[128, 128, … , 128]	[128, 256, … , 256, 128]
Hidden layer size (edge)	[128, 128, … , 128]	[64, 128, 256, … , 256]
Loss function	MSE	
Optimizer	Adam	
Learning rate scheduler	Lambda decay	
Lambda function	$(1 + k \cdot \lambda_0)^{-1}$	
Initial learning rate (λ_0)	5×10^{-5}	2×10^{-4}

the mass and momentum conversations after multiple iterations that sometimes can reach up to an order of 10,000. In contrast, the proposed neural network model directly predicts the flow fields that are closer to the convergence criteria of the governing equations, as illustrated in Fig. 1(b)(II). Hence, using the predicted solutions by GCN-based model to initialize the RANS solver will reduce the time taken for the convergence constraints to be met, resulting in a simulation that is both physically accurate and accelerated.

2.1. Neural network inputs and parameters

Table 1 outlines the neural network model parameters and training parameters used in the experiments. The input features can include node positions (2), SV (2), DID (8), inlet values (3), and freestream velocity and pressure fields (3) when residual training is applied. The eight inputs for DID are due to the choice of using eight segments, each spanning $\pi/4$ radians such that $\left[(-\pi/8, \pi/8), (\pi/8, 3\pi/8), \ldots, (13\pi/8, 15\pi/8)\right]$ represents the 360°-view from a given point. A segment that does not encounter the body in the flow is prescribed with a maximum distance value of $d_{max} = 5$. Hence, the input layer size for both networks is 15 and 18 without and with residual training, respectively. The output layer for all models consists of three features: the u, v and p fields. The mean squared error (MSE) loss function and Adam optimizer were utilized to train the neural networks, while a custom decay function was used for the learning rate. All models were trained using half-precision and distributed data parallelism across four GPUs.

2.2. Residual training

Residual training, which is extensively used in image super-resolution [31–34], involves utilizing differences between the ground truth and its estimate to ease the learning. Hence, in this case, instead of directly predicting the flow field φ_{GT}, the model is trained to predict the residual field by minimizing the following loss function:

$$\mathcal{L}\left(\varphi_{GT}, \widehat{\varphi} + \widetilde{\varphi}_{est}\right), \tag{1}$$

where $\widehat{\varphi}$ is the network output and $\widetilde{\varphi}_{est}$ represents an estimated flow field. Additionally, unlike conventional methods that depend on low-resolution simulations for initial estimates, this work used the freestream conditions as the starting point for residual training, thus keeping the cost to generate data for training to a minimum. Mathematically, this freestream approach can be expressed as:

$$\mathbf{U} = \mathbf{U}_{\infty} + \mathbf{U}', \tag{2}$$

where the overall flow velocity, $\mathbf{U}$, is the vector sum of the freestream conditions, $\mathbf{U}_{\infty}$, and the deviation from freestream, $\mathbf{U}'$, induced by the presence of a body within the flow and unique to its specific shape and orientation. Note that this decomposition is akin to the conversion from inertial frame to body frame and is also applicable to potential flows (i.e., superpositioning of inviscid freestream and singularity elements) and turbulent flows (i.e., steady mean and transient fluctuations) [35]. Readers can refer to Ref. [30] for further details on the residual training.

3. Datasets generation and validation

3.1. Types of datasets

This study focuses on the utilization of four-digit series airfoil shapes developed by the National Advisory Committee for Aeronautics (NACA) as the preferred choice for aircraft wing applications. The NACA airfoil series presents a wide variety of airfoil families, allowing for significant permutations between different types of airfoil shapes. Additionally, the ease of automating the mesh generation process further supports the selection of NACA airfoils for this investigation. The four-digit series NACA airfoils are parameterized using numerical codes ($MPXX$), where (i) the first digit (M) corresponds to the maximum camber as a percentage of the chord, (ii) the second digit (P) denotes the distance of maximum camber from the airfoil's leading edge in tenths of the chord, and (iii) the last two digits (XX) describe the maximum thickness of the airfoil as a percentage of the chord. This systematic parameterization of NACA airfoils allows for efficient exploration of a wide range of airfoil designs and facilitates the automation of the mesh generation.

Two-dimensional (2D) datasets were created in this work to develop the deep learning model in a wide variety of geometries and flow conditions. To examine the model performance in arbitrary geometries, all datasets comprise of symmetric and asymmetric airfoils for a comprehensive exploration of airfoil shapes. Also, for all datasets, the camber parameters, M and P, were uniformly chosen from the range [0, 6] and the thickness parameter, XX, was uniformly selected from [5, 25]. In general, the datasets can be divided into set A with a fixed low Reynolds number Re = 500 and an angle of attack (AoA) of $\alpha = 5°$, and set B with random high Re and AoA in the range of $Re \in [10^5, 5 \times 10^6]$ and $\alpha \in [-5°, 7°]$, respectively. In total, more than 2000 airfoil simulations were conducted (1014 and 1025 for datasets A and B, respectively). An 8 : 1 : 1 ratio was adopted to divide each dataset into training, validation, and testing sets. The details of the two datasets have been summarized in Table 2.

3.2. Meshing technique

Due to the high number of cases required for this study (more than 2000 in accordance with Sec. 3.1), an automatic mesh generation process was necessary for tractability and robustness, ensuring the meshes are both produced in a timely fashion and of sufficient quality. For all cases, the *blockMesh* utility tool provided by OpenFOAM-v2112 [36] was utilized to create a C-grid type of hexahedral mesh. The airfoil is situated at the center of the domain with a domain size of 20 chord lengths as shown in Fig. 2(a), whereas the near-field views of the airfoil are illustrated on the right of Fig. 2, showing representative (b) symmetric and (c) cambered NACA airfoils.

In dataset A, which considers Re = 500, a total number of 278 airfoil cells was used upon grid convergence study in Sec. 3.4 with both the leading edge, Δx_{LE}, and trailing edge, Δx_{TE} grid sizes equal to 5 mm. Since the Reynolds number is low Re the first cell thickness adjacent to the airfoil wall, Δx_{BL}, was set at 1 mm such that $y^+ \approx 1$ to fully resolve

Table 2
Description of datasets.

Airfoil dataset	M, P	XX	Re	α [°]	Sample size	Avg. cells
A, low Re	[0, 6]	[5, 25]	500	5	1014	123k
B, high Re	[0, 6]	[5, 25]	$[0.1, 5] \times 10^6$	[−5, 7]	1025	111k

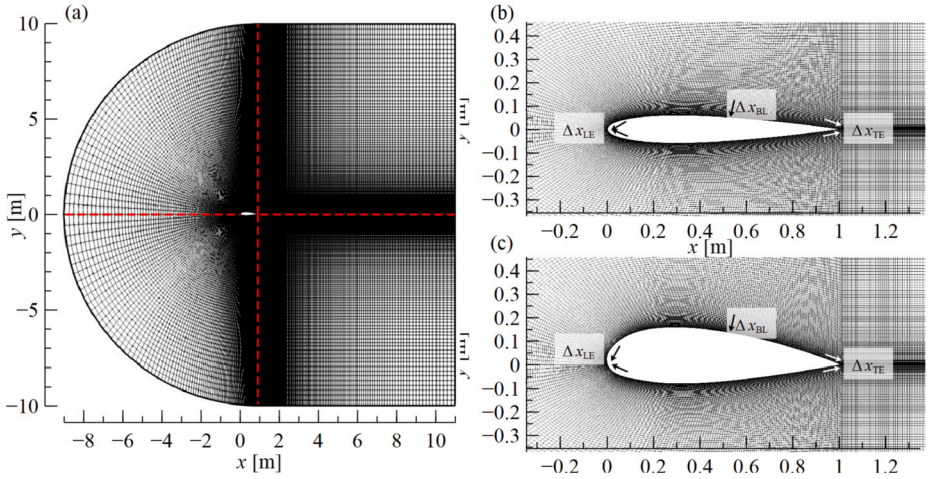

Fig. 2. (a) A type C-grid mesh was adopted where the airfoil is situated at the center of the domain. The near-field views are illustrated for representative (b) symmetric and (c) cambered NACA four-digit airfoils.

Table 3
Boundary conditions used for datasets A and B.

Field	Airfoil wall, $(\cdot)_{wall}$		Far-field, $(\cdot)_\infty$
	A	B	A and B
U	*noSlip*		*freestreamVelocity*
p	*zeroGradient*		*freestreamPressure*
k		*kLowReWallFunction*	
ω	*fixedValue*	*omegaWallFunction*	*freestream*
ν_t		*nutLowReWallFunction*	

Table 4
Discretization schemes chosen for dataset generation.

Operator	Discretization scheme
Gradient (∇)	Second-order Gauss linear
Divergence ($\nabla\cdot$)	Second-order upwind biased
Laplacian (∇^2)	Second-order Gauss linear
Interpolation	Second-order linear interpolation

the boundary layer around the airfoil geometry without adopting any wall function. Near the domain boundary, the far-field grid resolution, $\Delta x_{\text{far-field}} = 0.2$ m with a total expansion ratio of 40 along the outwards direction.

The same meshing method used in the dataset A was adopted for the dataset B, which considers a random and high Re flow condition. Each simulation in dataset B contains approximately a total number of 798 airfoil cells with $\Delta x_{\text{LE}} = \Delta x_{\text{TE}} = 0.2$ mm. $\Delta x_{\text{BL}} = 2 \times 10^{-5}$ m was prescribed to reduce the non-orthogonality of the mesh and ensure that $30 \leq y^+ \leq 100$ for the wall function to work efficiently.

3.3. Boundary conditions and solver settings

Table 3 tabulates the applied boundary conditions on the steady-state flow variables, namely the velocity vector, **U**, and kinematic pressure, p, which is divided by density, ρ, as well as the additional turbulent-related variables, such as turbulent kinetic energy, k, turbulent specific dissipation rate, ω, and turbulent kinematic viscosity, ν_t, which are needed for the k-ω shear stress transport (SST) model [37] used in this work. The turbulence model was employed for closure because of its suitability for aerodynamic flows with pressure-induced separation and adverse pressure gradient flows [38] and accessibility in OpenFOAM-v2112. No slip condition was used on the airfoil wall. Far-field boundaries refer to the outer domain boundary, for which the positive (outflow) and negative (inflow) fluxes use *zeroGradient* and *fixedValue*, respectively.

For low Re dataset A, the freestream velocity, $\mathbf{U}_\infty = [0.146,\ 0]$ m/s and kinematic viscosity, $\nu = 2.92 \times 10^{-4}$ m^2/s such that the Reynolds number, Re = 500. Gauge pressure was considered so the freestream pressure, $p'_\infty = 0$ m^2/s^2. The turbulent freestream values are chosen based on recommendations in [38]:

$$\nu_{t\infty} = 10^{-3}\nu\,, \quad \omega_\infty = \frac{C_\omega \mathbf{U}_\infty}{L}\,, \quad k_\infty = \nu_{t\infty}\omega_\infty\,,$$

where L is the approximate length of the computational domain, $C_\omega = 2$ for thin airfoil (i.e., $XX \leq 20$), and $C_\omega \geq 20$ for thick airfoil (i.e., $XX > 20$). For wall boundary condition, the following settings are recommended:

$$k_{\text{wall}} = 0\,, \quad \omega_{\text{wall}} = 10\frac{6\nu}{\beta_1 y_{\text{BL}}^2}\,,$$

where $\beta_1 = 0.075$. For dataset B, the $\mathbf{U}_\infty$ was calculated based on the Re, which is varying between 10^5 and 5×10^6 with $\nu = 1.461 \times 10^{-5}$ m^2/s and airfoil chord as the characteristic length. *omegaWallFunction* was used as wall function of the turbulent specific dissipation rate, ω, estimating y^+ by blending between the viscous and inertial sublayer with a binomial function [39]. The wall functions for turbulent kinetic energy, k, and turbulent viscosity, ν_t, are *kLowReWallFunction* and *nutLowReWallFunction*, respectively.

An incompressible steady-state flow solver, *simpleFoam*, was used to generate the A and B airfoil datasets. As implied, *simpleFoam* solves the finite-volume formulation of the steady, incompressible Navier–Stokes equations coupled with k-ω SST turbulence model using the SIMPLE algorithm [40]. The spatial discretization schemes used in the solver are summarized in Table 4. The resulting linear equations were solved using preconditioned (bi-)conjugate gradient (*PCG*) and stabilized preconditioned (bi-)conjugate gradient (*PBiCGStab*) with diagonal incomplete-Cholesky, (*DIC*), and diagonal incomplete LU, (*DILU*) as the preconditioner for symmetric and asymmetric matrices, respectively. The flow variables were updated with an under-relaxation factor of 0.3 for p and 0.5 for **U**, ω, and k, and converged to an order between 10^{-6} to 10^{-5}.

3.4. Validation

The grid convergence study on the low Re dataset A was conducted on three different levels of grid resolutions as tabulated in Table 5 with their corresponding lift and drag coefficients, c_l and c_d, respectively. Note that the thickness of the first cell adjacent to the airfoil wall was maintained at $y^+ < 1$ for all three grid resolutions. The c_l and c_d differences between the medium and fine cases are less than 0.5%, suggesting

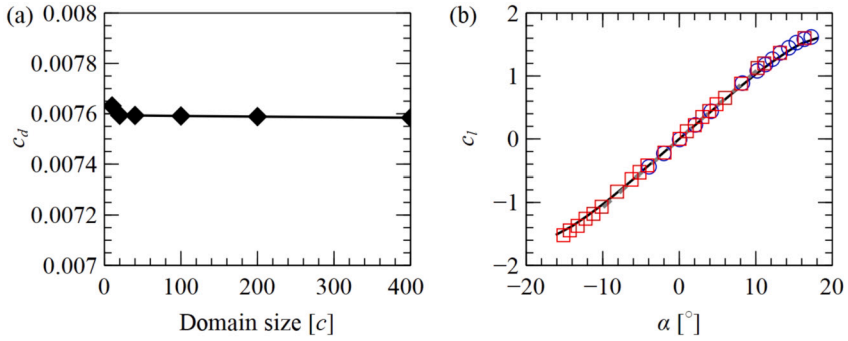

Fig. 3. Analyses of (a) domain independence and (b) accuracy for dataset B (NACA 0012, $\mathrm{Re} = 6 \times 10^6$, $\alpha = 0°$). In (b), the simulated c_l is given by a solid line, while its experimental counterparts are denoted by dotted-line [41], circles [42], and squares [43].

Table 5
Grid convergence analysis for dataset A (NACA 0012, $\mathrm{Re} = 500$, $\alpha = 5°$).

Resolution	Cell count	c_l	c_d
Coarse	30832	0.25881	0.18191
Medium	122952	0.26326	0.18179
Fine	512864	0.26254	0.18175

Table 6
Grid convergence analysis for dataset B (NACA 0012, $\mathrm{Re} = 6 \times 10^6$, $\alpha = 0°$).

Resolution	Cell count	c_d
Coarse	60876	0.007575
Medium	111036	0.007594
Fine	345280	0.007565

that the solutions are converged at the medium resolution. Thereafter, the grid resolution of the medium case was adopted to generate the entire single airfoil dataset A.

To account for the stronger wake effect in the high Re cases of dataset B, a domain independence study was first conducted with NACA 0012 airfoil at $\mathrm{Re} = 6 \times 10^6$ and $\alpha = 0°$ flow condition. The domain size of 20 chord lengths was found sufficient, as shown in Fig. 3(a). Thereafter, three different grid resolutions were considered for grid convergence analysis. The c_d results are tabulated in Table 6, indicating the medium grid resolution is sufficient. With the medium grid resolution, a comparison with experimental data [41–43] was conducted with AoA between −16° to 18°, as shown in Fig. 3(b). The simulation results agree with the measured data, especially between −10° to 10°, thus validating the simulation results from the medium resolution. As a result, the medium grid resolution was adopted for the entire dataset B.

4. Results and discussion

4.1. Low fixed Re and fixed AoA case

To demonstrate the effectiveness of introduced neural network (NN) components in Sec. 2, an ablation study was conducted as described in Table 7, building upon a baseline GCN model with geometry representations and/or residual training. Two different GCN architectures, namely MGN [21] and IVE [22], were considered to prove that the developed features are independent of NN architecture types.

The model performance was evaluated by the normalized mean squared errors (NMSE) averaged over all test samples, $\overline{\epsilon}_{\mathrm{NMSE}}$, where the NMSE of each test sample is computed from the differences between its flow variable predictions of $\mathbf{U}$ and p and their respective ground truths, normalized by a characteristic reference value, squared and averaged over all cell counts, and then averaged over the number of predicted flow variables. These steps are described in order by Eqn. (3):

Table 7
List of models with their integrated components.

Model	SV and DID	Residual training
Baseline	✗	✗
1	✓	✗
2	✓	✓

Table 8
Mean and standard deviation of average NMSE, $\overline{\epsilon}_{\mathrm{NMSE}}$, for all models of Table 7 of MGN and IVE architectures for dataset A. Note that all values are in 10^{-4} scale.

GCN	Baseline	1	2
MGN	6.25 ± 4.19	1.05 ± 0.53	$\mathbf{0.92 \pm 0.51}$
IVE	6.51 ± 4.01	0.92 ± 0.45	$\mathbf{0.71 \pm 0.44}$

$$\epsilon^{\varphi}_{\mathrm{NMSE},m} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{\varphi_{n,\mathrm{GT}} - \widehat{\varphi}_n}{\varphi_{\mathrm{ref}}}\right)^2\Bigg|_m , \tag{3a}$$

$$\epsilon_{\mathrm{NMSE},m} = \frac{1}{3}(\epsilon^{u}_{\mathrm{NMSE},m} + \epsilon^{v}_{\mathrm{NMSE},m} + \epsilon^{p}_{\mathrm{NMSE},m}) , \tag{3b}$$

$$\overline{\epsilon}_{\mathrm{NMSE}} = \frac{1}{M}\sum_{m=1}^{M}\epsilon_{\mathrm{NMSE},m} , \tag{3c}$$

where φ is a general notation for the u- or v-components of $\mathbf{U}$ or p, m is the counter for sample, n is the counter for cell, M and N denote the total number of samples and cells, respectively. The subscript $(\cdot)_{\mathrm{GT}}$ refers to the ground truth, while $\widehat{(\cdot)}$ indicates a prediction. The reference value, φ_{ref}, was freestream velocity magnitude, $\|\mathbf{U}_\infty\|$, and freestream dynamic pressure, $0.5\|\mathbf{U}_\infty\|^2$, for velocity components and pressure, respectively.

Table 8 presents the average NMSE, $\overline{\epsilon}_{\mathrm{NMSE}}$, of different models (see Table 7) for MGN and IVE architectures for dataset A. Regardless of GCN architectures, Model-1 with informed geometry representations (i.e., SV and DID) shows a significant improvement in field predictions of more than 80% in terms of $\overline{\epsilon}_{\mathrm{NMSE}}$ reduction compared to the baseline model. Similarly, improvements were observed for Model-2 with residual training, further lowering the $\overline{\epsilon}_{\mathrm{NMSE}}$ values for both MGN and IVE architectures. In other words, GCNs with both geometry representations and residual training incorporated have consistently achieved the lowest $\overline{\epsilon}_{\mathrm{NMSE}}$ compared to their baseline counterparts and those with only one component, thus confirming the efficacy of these components in improving GCN performance.

Given the superior performance of Model-2, Fig. 4 compares the performance of Model-2 across different GCN architectures using the box-and-whisker diagrams based on minimum/maximum range. IVE architecture consistently exhibits the lowest ϵ_{NMSE} as shown in Fig. 4(a). The breakdown of ϵ_{NMSE} into $\epsilon^{u}_{\mathrm{NMSE}}$, $\epsilon^{v}_{\mathrm{NMSE}}$, and $\epsilon^{p}_{\mathrm{NMSE}}$ for both ar-

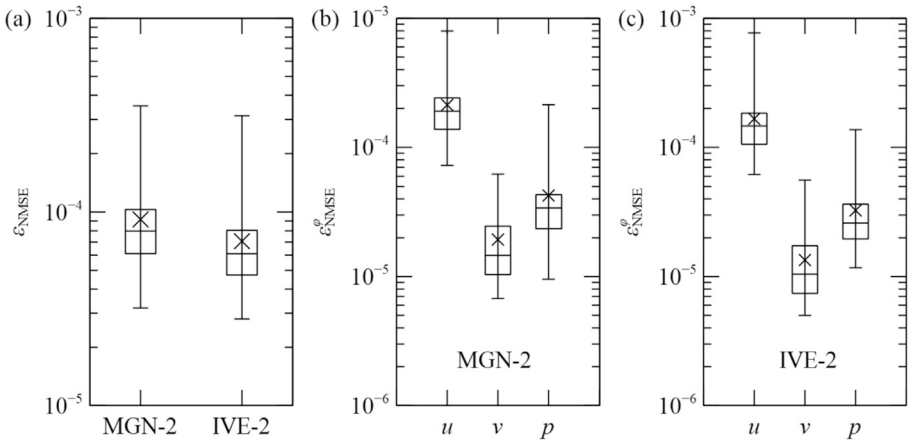

Fig. 4. Comparison of (a) ϵ_{NMSE} of Eq. (3b) between Model-2 of MGN and IVE architectures (i.e., MGN-2 and IVE-2, respectively), and the NMSE of each flow variable, $\epsilon^{\varphi}_{\text{NMSE}}$, of Eq. (3a) for (b) MGN-2 and (c) IVE-2 on dataset A. The cross marker represents the mean, while the line splitting the box in two represents the median. The lower and upper bounds of the box denote the first and third quartile, and the upper and lower ends the vertical line are the maximum and minimum values, respectively.

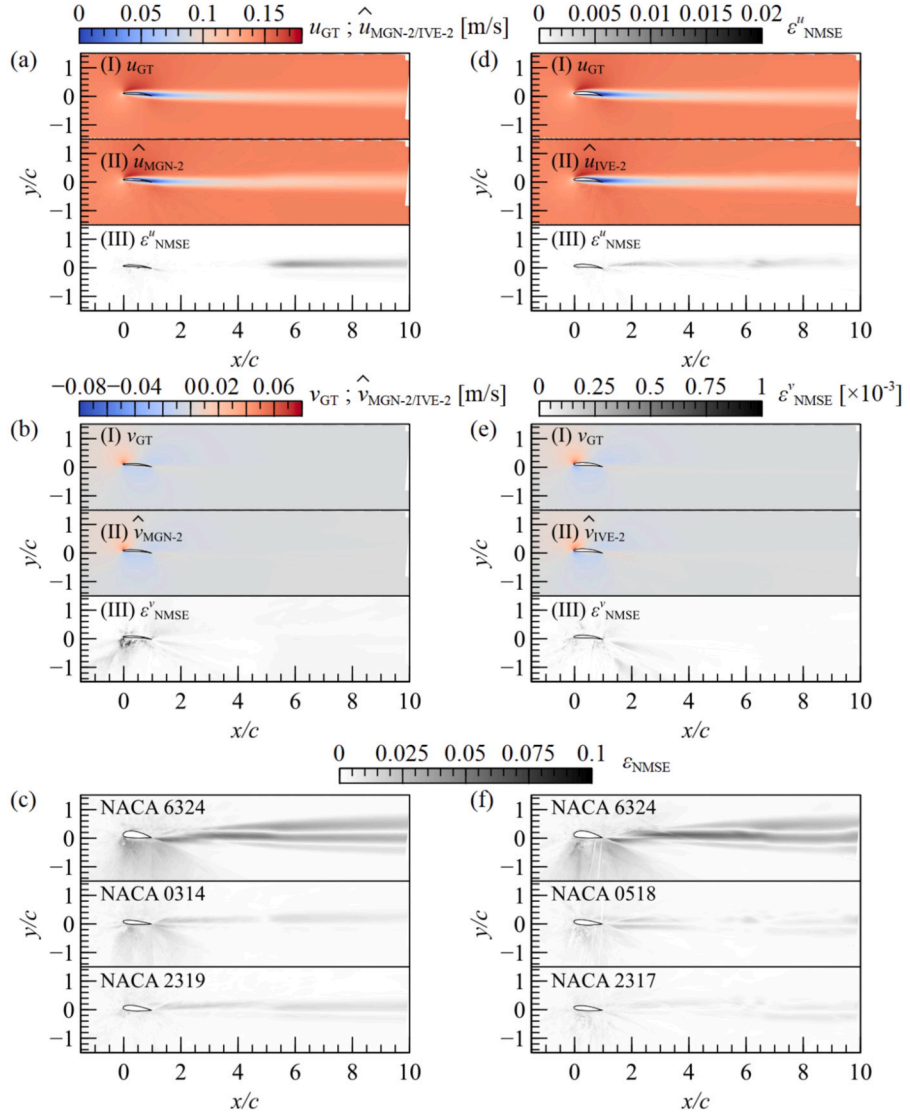

Fig. 5. Comparison of (top row) axial velocity component, u, and (middle row) transverse velocity component, v, obtained from (I) RANS simulation and (II) predictions by (left) MGN-2 or (right) IVE-2, as well as (III) their corresponding ϵ_{NMSE} contours for (a, b) NACA 4606 and (d, e) NACA 5610 test cases where $\epsilon_{\text{NMSE}} \approx \overline{\epsilon}_{\text{NMSE}}$ for MGN-2 and IVE-2, respectively. Contours for the maximum, median, and minimum ϵ_{NMSE} from (left) MGN-2 and (right) IVE-2 are shown at the bottom row (i.e., c, f) in descending order.

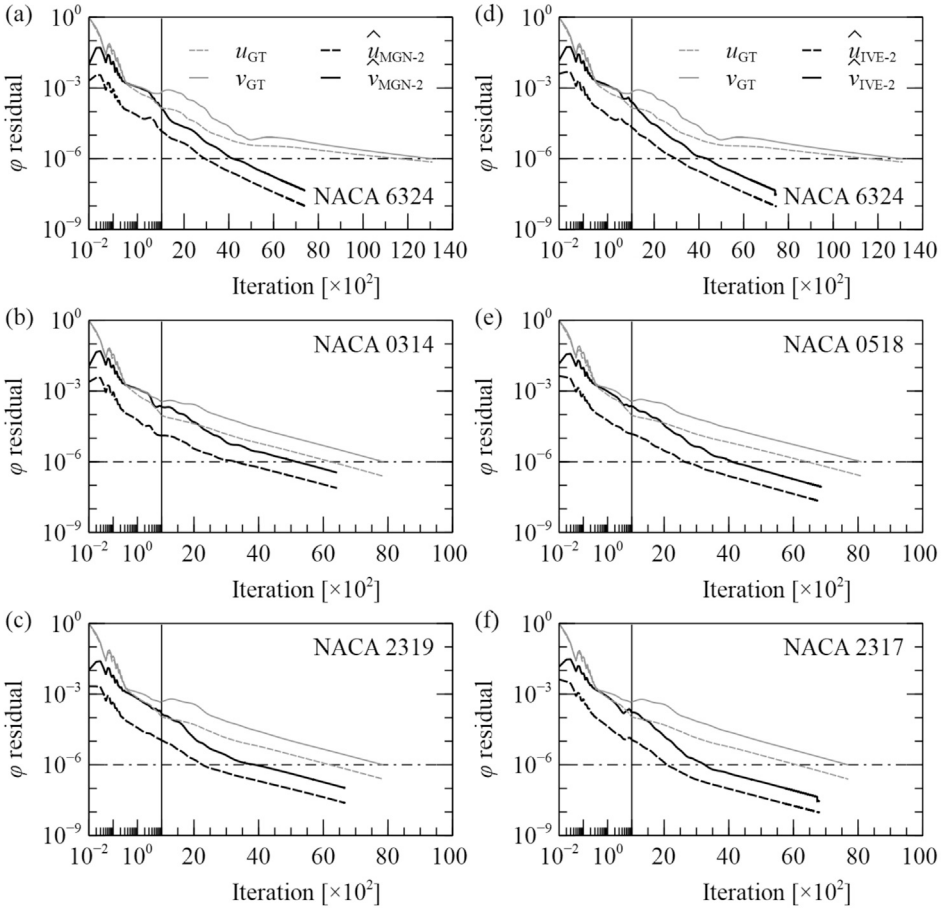

Fig. 6. Residual plots of (dashed lines) u- and (solid lines) v-components for (top row) maximum, (middle row) median, and (bottom row) minimum ϵ_{NMSE} cases (see Fig. 5(c)) initialized with predictions from (left column) MGN-2, $\widehat{(\cdot)}_{\text{MGN}-2}$, and (right column) IVE-2, $\widehat{(\cdot)}_{\text{IVE}-2}$, as well as the ground truths, $(\cdot)_{\text{GT}}$, obtained from initialization with freestream conditions. Part of the x-axis is plotted in logarithmic scale to highlight the lower initial residuals when initializing with NN predictions.

chitectures indicates that $\epsilon^{\varphi}_{\text{NMSE}}$ are in different orders where $\epsilon^{u}_{\text{NMSE}} > \epsilon^{p}_{\text{NMSE}} > \epsilon^{v}_{\text{NMSE}}$, as shown in Figs. 4(b) and (c), due to their corresponding magnitudes. As a result, the average NMSE, $\overline{\epsilon}_{\text{NMSE}}$, is expected to be dominated by the errors in u.

Two representative cases of NACA 4606 and NACA 5610 with $\epsilon_{\text{NMSE}} \approx \overline{\epsilon}_{\text{NMSE}}$ were selected to compare (I) ground truths, $(\cdot)_{\text{GT}}$, from RANS simulation and (II) predictions, $\widehat{(\cdot)}$, from MGN-2 and IVE-2, respectively, in Fig. 5. The comparison includes the contours of (a) axial velocity, u, and (b) transverse velocity, v. Additionally, the NMSE contours, ϵ_{NMSE}, for both velocity components are provided in (III) to highlight regions of high deviation. Apparently, the errors of u-field from both MGN-2 and IVE-2 are primarily concentrated in the near-field regions of the airfoil and downstream wake regions, at approximately 2% of $\|\mathbf{U}_{\infty}\|$. This concentration of errors can be attributed to the complex flow-body interactions and turbulence inherent to these regions, which, by conservation of momentum, will most affect the u-field in the flow over an airfoil problem. Conversely, the errors in the v-field are relatively smaller and generally localized around the airfoil, where v is most perturbed from their freestream conditions by the airfoil. Three additional $\epsilon^{u}_{\text{NMSE}}$ contours corresponding to the test cases with maximum, median, and minimum ϵ_{NMSE} values are presented in Fig. 5(c).

High errors are mostly observed in the wake regions, especially for the thicker airfoil, reinforcing the point that the average NMSE will largely be dominated by errors in the u-field. The results presented in Fig. 5 also suggest that the accuracy of GCN predictions will decline with increasing wake area where rapid changes in the flow and high gradients are prominent.

Despite their potential shortfall in cases of strong flow separation and/or large turbulent wake, the neural networks have demonstrated the ability to generate accurate predictions of flow fields for a wide range of airfoil shapes at Re = 500 and a fixed AoA of 5°. In the absolute sense, the variable-average error, ϵ_{NMSE}, for each cell is within the order of 10^{-3} as shown in Fig. 4. The predicted fields are also largely comparable to converged simulation results as illustrated in Fig. 5.

However, the resemblance to simulation results does not guarantee that the neural network predictions are physical, a condition that can be determined by checking the residuals of the governing equations computed with the predicted field variables. In this work, the check was performed by using the predicted fields as initial conditions for the same RANS solver that has simulated the ground truths. The hypothesis is that a prediction with sufficiently low residuals can theoretically meet the convergence criteria right away, thus fulfilling the physically accurate condition. A less optimal outcome is that the prediction's residuals are close, but not yet satisfy the convergence criteria, for which a reduction in iterations to convergence can still be expected, resulting in an accelerated simulation.

To confirm the above hypothesis, the residual plots for the maximum, median, and minimum ϵ_{NMSE} cases, same as Fig. 5(c), are presented in Fig. 6, comparing the convergence behavior of initialization with predictions from MGN-2 and IVE-2 to that with freestream conditions, from which the ground truths in earlier analyses of Figs. 4 and 5 were obtained. The residuals are plotted on a log scale to highlight the initial differences and the convergence behavior over iterations. Across all cases, both MGN-2 and IVE-2 exhibit significantly lower initial residuals compared to the freestream initialization. This is evident

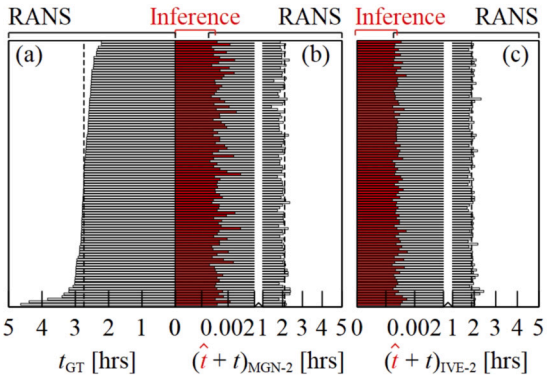

Fig. 7. Distribution of computational time taken of all test cases in dataset A. Time taken for RANS solver to achieve steady-state solutions with (a) freestream flow fields as initial conditions, t_{GT}, is shown on the left, while that with NN predictions is shown on the right for (b) MGN-2 and (c) IVE-2 architectures. Note that time taken for the predicted cases is a combination of the inference time by the NN and output time to the RANS solver, $\hat{t}_{MGN\text{-}2/IVE\text{-}2}$, which are indicated by the red bars, on top of the RANS simulation time, $t_{MGN\text{-}2/IVE\text{-}2}$. The vertical dashed line represents the median computational time.

from the log-scaled initial portion of the x-axis. These lower starting points suggest that the NN provide better initial guesses than freestream conditions. In addition, the higher gradient of residual curves by MGN-2 and IVE-2 demonstrates a sharper reduction in residual, leading to faster convergence. As supported by Fig. 6, a sufficiently accurate prediction can substantially reduce the computational effort required to meet convergence criteria, and even less optimal predictions still lead to accelerated simulations by lowering the starting residuals and reducing the number of required iterations. Interestingly, separate analyses (not shown here for brevity) indicate that the dominant factor for the acceleration of the simulations is the order-of-magnitude of the starting residuals and not their absolute value, which has not exhibited any clear trend with the total iteration count.

The computational time required for all 102 test cases in dataset A is depicted in Fig. 7 to quantify the time saving attained from the faster convergence shown in Fig. 6. The left bars in the figure represent the time taken for the RANS solver to achieve steady-state solutions with freestream flow fields as initial conditions, t_{GT}. The right red bars show $\hat{t}_{MGN\text{-}2/IVE\text{-}2}$, the time taken for the neural networks to perform inference tasks and output initialization files that are compatible with the RANS solver, which is generally trivial as compared to the time taken for the RANS solver to converge to steady state from the neural network predictions. Nonetheless, total computational time of neural network cases, given by $(\hat{t}+t)_{MGN\text{-}2/IVE\text{-}2}$, is presented for comprehensiveness, necessitating a scale break in the x-axis of Fig. 7.

Based on Fig. 7, the time taken for the RANS solver to achieve steady-state solutions using the initial conditions from the MGN-2/IVE-2 model is significantly reduced compared to that starting with freestream flow fields. Quantitatively, the median t_{GT} is approximately 2.74 hours. In contrast, the combined times for the MGN-2 and IVE-2 models, $(\hat{t}+t)_{MGN\text{-}2/IVE\text{-}2}$, are significantly lower with median at approximately 2.02 and 1.93 hours, respectively, and maximum approximately equal to the median t_{GT}. This reduction in computational time demonstrates the efficiency of both models, particularly IVE-2 in providing accurate initial conditions that expedite the convergence process of the RANS solver.

The distribution of percentage time reduction by using predictions from MGN-2 and IVE-2 as initial conditions is illustrated in Fig. 8 for the 102 test cases presented in Fig. 7. Overall, all test cases have achieved computational time reduction for the steady-state solutions (i.e., > 0%), highlighting the effectiveness of both MGN-2 and IVE-2 architectures. Specifically, the more common percentage time reduction occurs at approximately 20% – 30% and 20% – 40% ranges for MGN-2 and IVE-2, respectively. Both architectures also exhibit time reduction as high as 60%. These findings underscore the potential of integrating GCN, regardless of its architecture, with a numerical fluid solver to enhance its efficiency. Note that time reduction has also been observed with different initial conditions that are commonly used in CFD, as shown by the analyses in Appendix A.

4.2. High variable Re and variable AoA case

In the following, dataset B that pertains to 103 random test cases of airfoils in flow of $10^5 \leq \mathrm{Re} \leq 5\times10^6$ and variable AoA between the range of $-5°$ to $7°$ is considered. For brevity but without loss of generality, only IVE-2 results will be presented because, as found in Sec. 4.1, both MGN-2 and IVE-2 return shorter computational time when their predictions were used to initialize the RANS solver than the conventional approach where the freestream conditions were applied for initialization, while IVE-2 has slightly better performance over MGN-2.

A representative case of NACA 5114 with ϵ_{NMSE} approximately equal to the median of ϵ_{NMSE} of Eq. (3b) was selected to compare (I) ground truths, $(\cdot)_{GT}$, from RANS simulation and (II) predictions, $\widehat{(\cdot)}$, from IVE-2 in Fig. 9. The comparison includes the contours of (a) axial velocity, u, (b) transverse velocity, v, and (c) kinematic gauge pressure, p. Additionally, the NMSE contours, ϵ_{NMSE}, for velocity and pressure are provided in (III) to highlight regions of high deviation. The velocity field of the ground truth and prediction agree in general, with relatively low $\epsilon^{\varphi}_{NMSE}$ everywhere except in wake regions, most noticeable in the u-component ϵ_{NMSE} plot. In contrast, the predicted pressure, $\hat{p}$, shows more significant deviations from its corresponding ground truth around the near fields of the airfoil, suggesting that pressure-velocity coupling is not currently observed by the NN method.

Following the convention of Fig. 5, the NMSE contours of three test cases with the maximum, median, and minimum ϵ_{NMSE} values corresponding to NACA 2317 ($\mathrm{Re} \sim 10^5$), NACA 5114 ($\mathrm{Re} \sim 10^6$), and NACA 2517 ($\mathrm{Re} \sim 10^6$), respectively, are shown in Fig. 9(d). The increment of predictive performance at higher Re can be attributed to the fact that

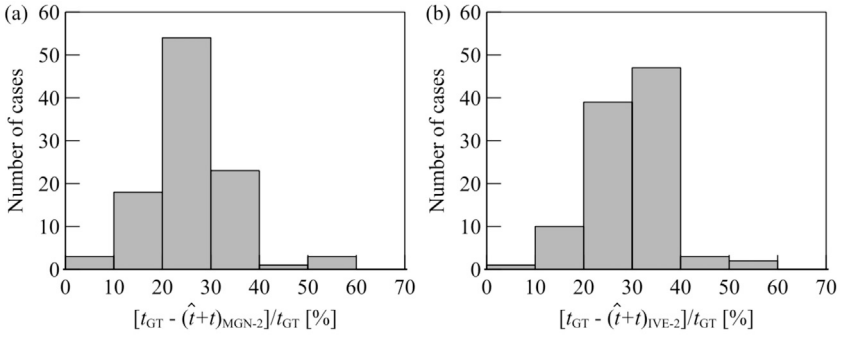

Fig. 8. Reduction in computational time by initialization with predictions from (a) MGN-2 and (b) IVE-2 with respect to ground truth simulations initialized with freestream conditions for dataset A.

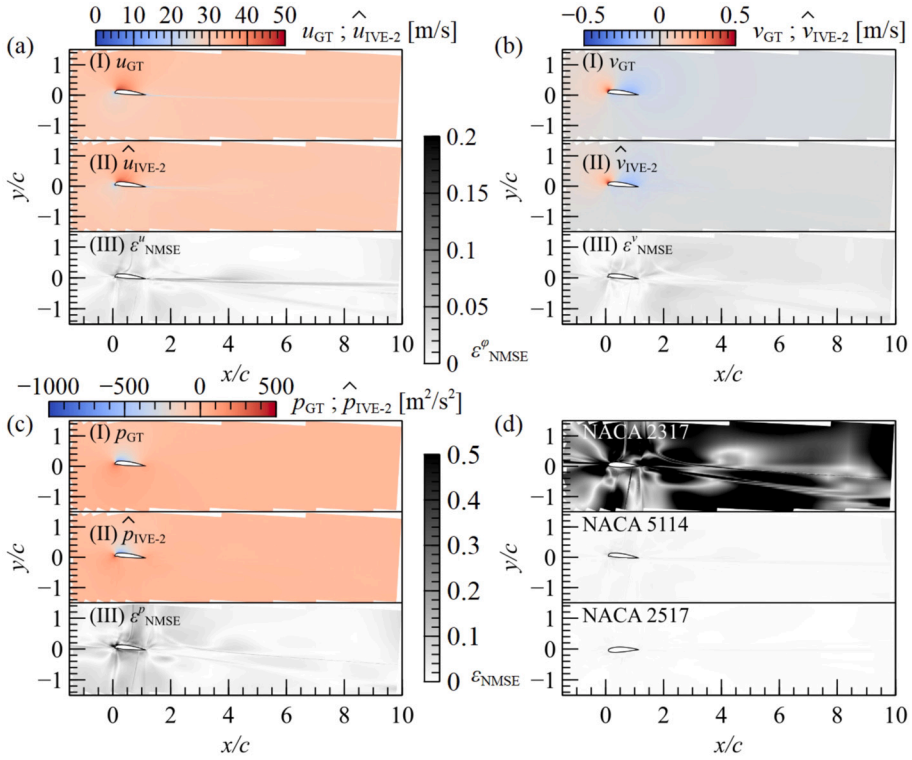

Fig. 9. Comparison of (a) axial velocity component, u, (b) transverse velocity component, v, and (c) kinematic gauge pressure, p, obtained from (I) RANS simulation and (II) predictions by IVE-2, as well as (III) their corresponding ϵ_{NMSE} contours for NACA 5114 with median ϵ_{NMSE} value. Contours for the maximum, median, and minimum ϵ_{NMSE} are shown in (d) in descending order.

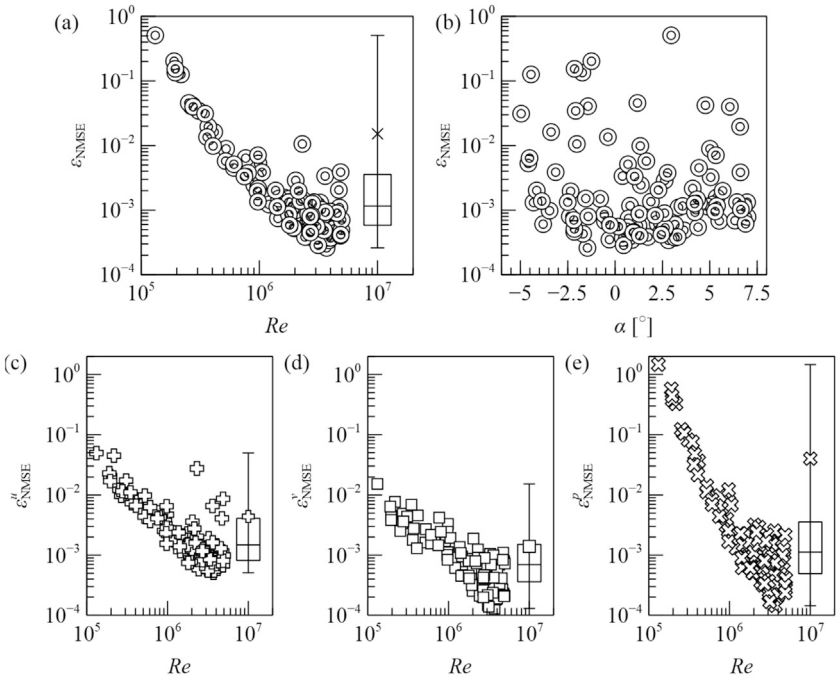

Fig. 10. ϵ_{NMSE} dependencies on (a) Reynolds number, Re, and (b) angle of attack, α, for dataset B under IVE-2 model predictions. Each data point represents a test case (total 103 cases) at a specific flow condition. The (c) u, (d) v, and (e) p flow fields NMSE, $\epsilon^{\varphi}_{NMSE}$ distributions under the influence of Re. The box plots represent the statistical mean (markers), median (lines that split the boxs), 1st (lower bound) and 3rd (upper bound) quartiles of ϵ_{NMSE} values.

higher Re tends to result in a more stable turbulent boundary layer with less pronounced separation, which is easier for the NN to predict accurately.

This trend is corroborated in Fig. 10 which depicts the correlation between ϵ_{NMSE} and (a) Re and (b) AoA, α, as well as (c–e) $\epsilon^{\varphi}_{NMSE}$ of Eq. (3a) and Re. In general, ϵ_{NMSE} is seen to decrease with increasing Re, justifying the observation in Fig. 9. In contrast, no clear correlation between ϵ_{NMSE} and AoA is observed. Further analyses of the $\epsilon^{\varphi}_{NMSE}$ of individual flow fields u, v, and p reveal a similar trend to the overall ϵ_{NMSE} where errors are lower at higher Re. The box plots in Figs. 10(c)

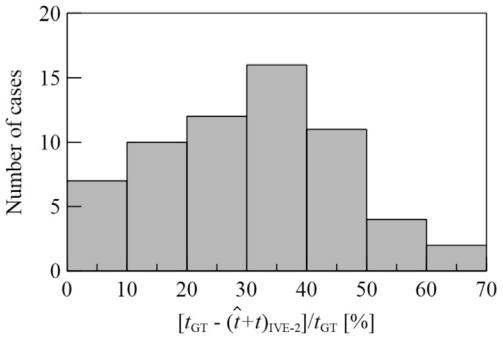

Fig. 11. Reduction in computational time by initialization with predictions from IVE-2 relative to ground truth simulations initialized with freestream conditions for dataset B.

– (e) provide a detailed statistical summary of the ϵ_{NMSE} distribution where the high ϵ_{NMSE} in lower Re test cases lead to large deviation between mean and median ϵ_{NMSE} values. In short, the IVE-2 is well-suited for steady or mildly unsteady flows with limited wake turbulence and separation, though incorporating more low-Re cases into the training dataset could further enhance performance.

The time reduction achieved by using IVE-2 predictions as initial conditions is quantified in Fig. 11. Like the low Re, fixed AoA cases evaluated in Sec. 4.1, the results demonstrate that significant computational time savings can be attained for high Re and variable AoA scenarios too. Specifically, approximately 75% of the test cases benefit from time savings, with a substantial number of cases exhibiting reductions between 20% and 50% and a peak around 40%. This analysis confirms that NN predictions are capable of accelerating RANS simulations for a wide range of Re from order as low as 10^2 to as high as 10^6 and mild AoA within the generally pre-stall regime.

5. Conclusion

In this work, a novel approach using graph neural network predictions for flow field initialization to accelerate Reynolds-averaged Navier–Stokes (RANS) simulations is presented. Two neural network architectures, namely MeshGraphNet (MGN) [21] and Invariant Edge-Graph Convolutional Networks (IVE) [22], were employed to demonstrate the general applicability of this deep learning approach. The neural networks were trained using residual training techniques and with the incorporation of geometric representation via shortest vector and directional integrated distance [30] between any cell of the computational domain and the body boundary inside the flow.

The performance of the neural network approach was evaluated using two distinct datasets, one that focuses on fixed low Reynolds number of 500 and angle of attack of 5° and the other corresponding to high Reynolds numbers in order of 10^6 and angle of attack ranging from −5° to 7°. The results from both datasets consistently demonstrate the effectiveness of the neural network approach in achieving high prediction accuracy that can significantly reduce computational time by up to 60% and 70% for low and high Reynolds numbers, respectively. Since the improvements were observed for both MGN and IVE and are of qualitatively similar trend between the two datasets, the findings are reasonably shown to be neural network architecture-agnostic.

Overall, this work has showcased the potential of integrating neural network into a traditional steady RANS solver to enhance its efficiency in computational fluid dynamics applications, especially for the aeronautical engineering sector. While still limited by their time independence and modeling of the entire turbulence spectrum, these findings pave the way for future research to further optimize deep learning models in fluid simulations. A good starting point could be the incorporation of continuity constraint in incompressible flow predictions. With more of such physics knowledge included, neural network-computational fluid dynamics will no doubt be increasingly generalized. In turn, the approach can be extended for efficient multi-disciplinary design and optimization space exploration, and not only on single-body scenarios like an airfoil or a wing, but also cases with multiple repeated/similar units such as tandem-wing aircraft [44] and compressor cascade [45]. For the latter, Pan et al. [46] has recently demonstrated that even unsteady RANS is not sufficient to capture the configuration's full dynamics, suggesting that the neural network also has to be compatible with higher-fidelity methodology like large-eddy simulation. In doing so, other flow problems involving such complexities as chemical reactions and combustion, compressible effects, three-dimensionality, and unsteady turbulence can eventually benefit from the wave of artificial intelligence for fluid science.

CRediT authorship contribution statement

Wei Xian Lim: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Loh Sher En Jessica:** Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Yu Lv:** Writing – review & editing, Supervision. **Adams Wai-Kin Kong:** Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **Wai Lee Chan:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Conceptualization.

Funding sources

This study is supported under the RIE2020 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) Funding Initiative (Project # 11801E0033), as well as cash and in-kind contributions from Rolls-Royce Singapore Pte Ltd.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Wai Lee Chan, Loh Sher En Jessica, and Adams Wai-Kin Kong have patent #GB2623618 - Fluid flow simulation issued to Rolls-Royce PLC. 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.

Acknowledgements

The computational work for this article was partially performed on resources of the High-Performance Computing Center of NTU, Singapore (https://www.ntu.edu.sg/hpcc), and National Super computing Center, Singapore (https://www.nscc.sg). The corresponding author gratefully acknowledges the support by the Chinese Academy of Sciences (CAS) President's International Fellowship Initiative (2023VMC0022), which has enabled his intellectual exchanges at CAS that greatly benefited this work.

Appendix A. Comparison of initial conditions (dataset A)

The analysis encompasses three different initial conditions: freestream, null, and potential flow estimated conditions for dataset A. In the left column, Fig. A.1 shows the time reduction obtained with the MGN-2 model, while the right column presents those obtained with the IVE-2 model. The top, middle, and bottom rows show the computational time reduction relative to initialization with freestream (i.e., $u = \|\mathbf{U}_\infty\|$, $v = 0$), $(\cdot)_{GT}$, null (i.e., $u = v = 0$), $(\cdot)_{GT0}$, and potential flow conditions, $(\cdot)_{PoF}$.

For the freestream initial conditions, the MGN-2 model shows that the majority of test cases exhibit time reductions between 30% and 40%, with some test cases achieving up to 90% reduction. The IVE-2 model

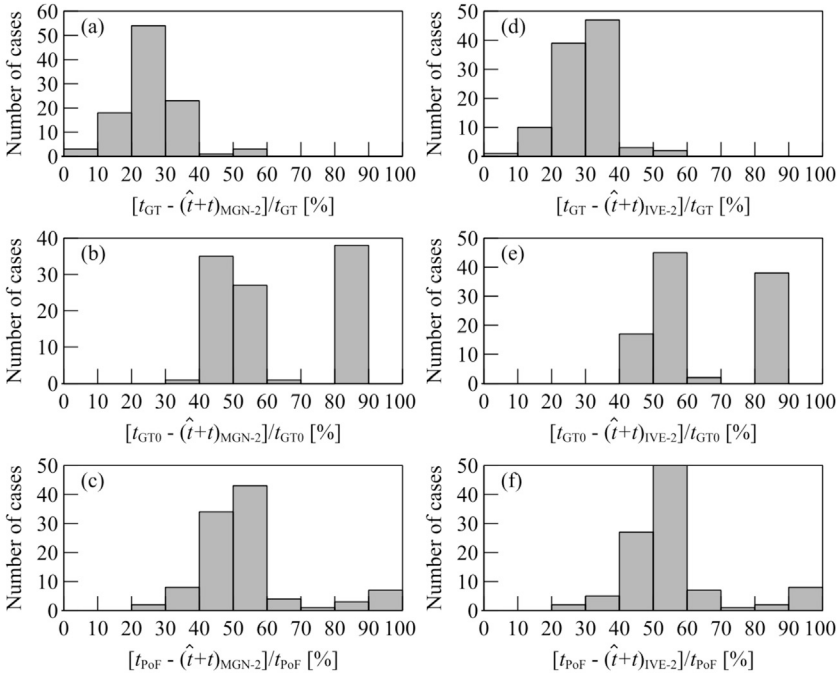

Fig. A.1. Reduction in computational time by initialization with predictions from (left column) MGN-2 and (right column) IVE-2 with respect to initializing with (top row) freestream (repeated from Fig. 8), $(\cdot)_{GT}$, (middle row) null (i.e., $u = v = 0$), $(\cdot)_{GT0}$, and (bottom row) potential flow conditions, $(\cdot)_{PoF}$, for dataset A.

similarly shows a concentration of cases with time reductions between 30% and 40%, but with a more uniform distribution extending up to 90%.

When comparing against null initial conditions, both MGN-2 and IVE-2 models demonstrate a bimodal distribution, with a significant number of cases clustering around the ranges of 30% – 40% and 70% – 80%. The IVE-2 model appears to offer a slightly more consistent performance, as evidenced by the dense clustering of points within these two ranges.

The potential flow initial conditions reveal a broader distribution of time reductions for both models. The MGN-2 model shows a spread with reductions primarily between 30% and 70%, while the IVE-2 model demonstrates a more concentrated performance, with a notable clustering around 40% to 60% reduction.

Data availability

The airfoil datasets are publicly accessible at https://doi.org/10.21979/N9/9OYSTD.

References

[1] N.J. Cherry, R. Hillier, M. Latour, Unsteady measurements in a separated and reattaching flow, J. Fluid Mech. 144 (1984) 13–46.
[2] M. Errera, A. Dugeai, P. Girodroux-Lavigne, J.D. Garaud, M. Poinot, S. Cerqueira, G. Chaineray, Multi-physics coupling approaches for aerospace numerical simulations, Aerosp. Lab 2 (2011) ALS02–09.
[3] P.R. Spalart, V. Venkatakrishnan, On the role and challenges of CFD in the aerospace industry, Aeronaut. J. 120 (2016) 209–232.
[4] F. Moukalled, L. Mangani, M. Darwish, F. Moukalled, L. Mangani, M. Darwish, The Finite Volume Method, Springer, 2016.
[5] J. Tu, G.H. Yeoh, C. Liu, Y. Tao, Computational Fluid Dynamics: a Practical Approach, Elsevier, 2023.
[6] S. Lawson, M. Woodgate, R. Steijl, G. Barakos, High performance computing for challenging problems in computational fluid dynamics, Prog. Aerosp. Sci. 52 (2012) 19–29.
[7] J.P. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, D.J. Mavriplis, CFD vision 2030 study: a path to revolutionary computational aerosciences, Technical Report, National Aeronautics and Space Administration (NASA), 2014.
[8] H.C. Chen, V.C. Patel, S. Ju, Solutions of Reynolds-averaged Navier–Stokes equations for three-dimensional incompressible flows, J. Comput. Phys. 88 (1990) 305–336.
[9] J.W. Deardorff, A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers, J. Fluid Mech. 41 (1970) 453–480.
[10] D.J. Lucia, P.S. Beran, W.A. Silva, Reduced-order modeling: new approaches for computational physics, Prog. Aerosp. Sci. 40 (2004) 51–117.
[11] D. Schmidt, R. Maulik, K. Lyras, Machine learning accelerated turbulence modeling of transient flashing jets, Phys. Fluids 33 (2021).
[12] K. Li, J. Kou, W. Zhang, Unsteady aerodynamic reduced-order modeling based on machine learning across multiple airfoils, Aerosp. Sci. Technol. 119 (2021) 107173.
[13] V. Rozov, C. Breitsamter, Data-driven prediction of unsteady pressure distributions based on deep learning, J. Fluids Struct. 104 (2021) 103316.
[14] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[15] D. Kochkov, J.A. Smith, A. Alieva, Q. Wang, M.P. Brenner, S. Hoyer, Machine learning–accelerated computational fluid dynamics, PNAS 118 (2021) e2101784118.
[16] M. Aliakbari, M. Mahmoudi, P. Vadasz, A. Arzani, Predicting high-fidelity multiphysics data from low-fidelity fluid flow and transport solvers using physics-informed neural networks, Int. J. Heat Fluid Flow 96 (2022) 109002.
[17] O. Obiols-Sales, A. Vishnu, N. Malaya, A. Chandramowliswharan, CFDNet: a deep learning-based accelerator for fluid simulations, in: Proceedings of the 34th ACM International Conference on Supercomputing, ICS '20, 2020.
[18] X. Hui, J. Bai, H. Wang, Y. Zhang, Fast pressure distribution prediction of airfoils using deep learning, Aerosp. Sci. Technol. 105 (2020) 105949.
[19] J.P. Romano II, Faster, cheaper, and better CFD: a case for machine learning to augment Reynolds-averaged Navier–Stokes, Ph.D. thesis, Old Dominion University, 2023.
[20] F.D.A. Belbute-Peres, T. Economon, Z. Kolter, Combining differentiable PDE solvers and graph neural networks for fluid flow prediction, Proc. Mach. Learn. Res. 119 (2020) 2402–2411.
[21] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, P. Battaglia, Learning to simulate complex physics with graph networks, Proc. Mach. Learn. Res. 119 (2020) 8459–8468.
[22] J. Chen, E. Hachem, J. Viquerat, Graph neural networks for laminar flow prediction around random two-dimensional shapes, Phys. Fluids 33 (2021) 123607.
[23] F. Ogoke, K. Meidani, A. Hashemi, A.B. Farimani, Graph convolutional networks applied to unstructured flow field data, Mach. Learn.: Sci. Technol. 2 (2021) 045020.
[24] F. Bonnet, J. Mazari, P. Cinnella, P. Gallinari, AirfRANS: high fidelity computational fluid dynamics dataset for approximating Reynolds-averaged Navier–Stokes solutions, Adv. Neural Inf. Process. Syst. 35 (2022) 23463–23478.
[25] F. Meire, P. Tobias, W. Peter, P. Alexander, B. Peter, Multiscale meshgraphnets, in: 2nd AI4Science Workshop at the 39th International Conference on Machine Learning, 2022, https://openreview.net/pdf/a7bf090f67f8d2b4d254d94296645ac54ac81d6d.pdf.
[26] M. Lino, S. Fotiadis, A.A. Bharath, C. Cantwell, Towards fast simulation of environmental fluid mechanics with multi-scale graph neural networks, in: Workshop

on AI for Earth and Space Science at the 10th International Conference on Learning Representations, 2022, https://ai4earthscience.github.io/iclr-2022-workshop/camera_ready/iclr_2022_ai4ess_14.pdf.
[27] S. Strönisch, M. Meyer, C. Lehmann, Flow field prediction on large variable sized 2D point clouds with graph convolution, in: Proceedings of the Platform for Advanced Scientific Computing Conference, 2022.
[28] Y. Cao, M. Chai, M. Li, C. Jiang, Efficient learning of mesh-based physical simulation with bi-stride multi-scale graph neural network, Proc. Mach. Learn. Res. 202 (2023) 3541–3558.
[29] X. Peng, X. Li, X. Chen, X. Chen, W. Yao, A hybrid deep learning framework for unsteady periodic flow field reconstruction based on frequency and residual learning, Aerosp. Sci. Technol. 141 (2023) 108539.
[30] S. Loh, N. Arafat, W. Lim, W. Chan, W. Kong, Finite volume features, global geometry representations, and residual training for deep learning-based CFD simulation, Proc. Mach. Learn. Res. 235 (2024) 21555–21576.
[31] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 136–144.
[32] Y. Tai, J. Yang, X. Liu, Image super-resolution via deep recursive residual network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3147–3155.
[33] J. Li, F. Fang, K. Mei, G. Zhang, Multi-scale residual network for image super-resolution, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 517–532.
[34] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu, Residual dense network for image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2472–2481.
[35] C.G. Speziale, A review of material frame-indifference in mechanics, Appl. Mech. Rev. 51 (1998) 489–504.
[36] H. Jasak, A. Jemcov, Z. Tukovic, OpenFOAM: a C++ library for complex physics simulations, in: International Workshop on Coupled Methods in Numerical Dynamics, vol. 1000, 2007.
[37] F.R. Menter, M. Kuntz, R. Langtry, Ten years of industrial experience with the SST turbulence model, in: Turbulence, Heat and Mass Transfer 4: Proceedings of the Fourth International Symposium on Turbulence, Heat and Mass Transfer, Begell House, 2003, pp. 625–632.
[38] F.R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA J. 32 (1994) 1598–1605.
[39] F.R. Menter, T. Esch, Elements of industrial heat transfer predictions, in: Proceedings of Congresso Brasileiro de Engenharia Mecânica, 2001, pp. 117–127.
[40] L. Caretto, A.D. Gosman, S.V. Patankar, D.B. Spalding, Two calculation procedures for steady, three-dimensional flows with recirculation, in: Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics: Vol. II Problems of Fluid Mechanics, 1973, pp. 60–68.
[41] W.J. McCroskey, A critical assessment of wind tunnel results for the NACA 0012 airfoil, Technical Report USAAVSCOM TR 87-A-5, NASA Ames Research Center, 1987.
[42] C.L. Ladson, Effects of independent variation of Mach and Reynolds numbers on the low-speed aerodynamic characteristics of the NACA 0012 airfoil section, Technical Report NASA TM-4074, NASA Langley Research Center, 1988.
[43] I.H. Abbott, A.E. Von Doenhoff, Theory of Wing Sections: Including a Summary of Airfoil Data, Dover Publications, 2012.
[44] B. Yin, Y. Guan, A. Wen, N. Karimi, M.H. Doranehgard, Numerical simulations of ultra-low-Re flow around two tandem airfoils in ground effect: isothermal and heated conditions, J. Therm. Anal. Calorim. 145 (2021) 2063–2079.
[45] P.T. Hedje, L. Bricteux, Y. Bechane, S. Lavagnoli, Large eddy simulations of a high-speed low-pressure turbine cascade at subsonic and transonic Mach numbers, J. Turbomach. 147 (2025) 101010.
[46] T. Pan, T. Li, Z. Yan, Q. Li, Investigation of turbulence-induced disturbances and their evolution to stall onset in a compressor cascade using large eddy simulation, Chin. J. Aeronaut. (2025) 103491, https://doi.org/10.1016/j.cja.2025.103491, in press.