Generalizability of Convolutional Encoder-Decoder Networks for Aerodynamic Flow-field Prediction Across Geometric and Physical- Fluidic Variations

Kaustubh Tangsali
Graduate Research Assistant
J. Mike Walker '66 Department
of Mechanical Engineering
Texas A&M University
College Station, Texas 77843
Email: kmtangsali@tamu.edu

Vinayak R. Krishnamurthy
Assistant Professor
J. Mike Walker '66 Department
of Mechanical Engineering
Texas A&M University
College Station, Texas 77843
Email: vinayak@tamu.edu

Zohaib Hasnain*
Research Assistant Professor
J. Mike Walker '66 Department
of Mechanical Engineering
Texas A&M University
College Station, Texas 77843
Email: zhasnain@tamu.edu

The generalizability of a convolutional encoder-decoder based model in predicting aerodynamic flow field across various flow regimes and geometric variation is assessed. A rich master dataset consisting of 11,000+ simulations including cambered, uncambered, thin and thick airfoils simulated at varying angles of attack is generated. The various Mach and Reynolds number (Re) chosen allows analysis across compressible, incompressible, low and high Re flow regimes. Multiple studies are carried out with the model trained on datasets that are categorized based on the above parameters. In each study, the loss of prediction accuracy by training the model on a larger dataset (generalizability), versus a smaller categorically sorted dataset, is evaluated. Largely disparate flow features across the Re range lead to a 25.56% loss, while the generalization across Mach range led to an average of 23.95% loss. However, flow-field changes induced due to geometric variation exhibited a better generalization potential, through an increased accuracy of 12.4%.The encoder-decoder architecture allows extraction of relevant geometric features from largely different geometries (geometric generalization) providing a better out-of-sample prediction accuracy in comparison to physics-based generalization. It is shown that, through user-informed choice of training data (removal of geometrically similar samples), computational costs incurred in generating training data can be reduced. This is important for the application of such methods in the design optimization of platforms and components that require analysis of the fluid flows.

1 Introduction

Recently, there has been an increased interest towards the application of data-driven techniques to fluidic problems that previously required a rigorous solution of complex Partial Differential Equations either computationally or empirically. Data-driven techniques have the potential to revolutionize the field of fluid mechanics by cutting down the computation time by several orders of magnitude. This has led to these methods making their way into the design space exploration, optimization and control problems at an increasing pace [1]. However the question of generalizability, specifically, in the context of fluidic prediction for applied design of aerospace systems, remains unanswered. This domain is particularly challenging with respect to geometry (e.g. bluff bodies such as the Apollo capsules to streamlined fighter aircraft) and the flow physics (e.g. Mach and Reynolds numbers that encompass incompressible to reacting flows). In this paper, we present a detailed study of the generalizability of convolutional neural networks, specifically an encoder-decoder architecture, to these two broad design variables: geometry and flow physics.

This research is motivated by the immense success of deep learning approaches witnessed by the computer-aided geometric design community in the recent past. However,

*Address all correspondence to this author.

unlike their application to geometric design synthesis [2–4], the use of data-driven learning, especially neural network based approaches, to fluids problems is yet to mature toward full scale application to design synthesis. Even though there is a considerable effort on statistical and data driven approaches for flow prediction, significant work needs to be done toward creating a sound methodology.

1.1 Background

There are two broad classes of data-driven modeling approaches for fluid-related problems. In the first category, learning algorithms are used to improve the accuracy of the lower order mathematical and computational fluid dynamics (CFD) models. This includes the use of learning algorithms to providing better turbulence closure [5–7], improved discretization [8], dimensionality reduction [9] and data-driven solving and identification of partial differential equations [10, 11]. These methods use machine learning techniques to supplement established methods and laws by improving their accuracy or reducing the computational cost.

The second category of efforts tries to leverage the generalization capability of the learning algorithms. Such methods, when trained with sufficiently large pairs of input-output combinations, can generate the results for out-of-sample input data. Deep learning based models are especially suited for this type of application as their power to generalize is established in various image processing applications [12]. Kutz [13] summarizes various approaches of implementing the Deep Neural Networks to fluid flow problems. One such special type of Deep Neural Networks (DNN) is the type of networks that implements the Convolutional operation. These networks are especially suited for the matrix representations of the data and have the capability to reduce the trainable parameters by weights sharing. It is also found that the generalization capability of such networks is better than other types of DNN [12]. Works found in literature show the application of this convolution operation to predict flow fields [14–20], specific flow quantities [21–23], flow initialization [24] and solving inverse design problems [25, 26]. Other DNN based studies which are more focused on the fluid simulations for computer graphics applications can be found in [27, 28].

Guo et al. [14] initially explored the possibility for studying the low Re flow over complicated car geometries using just simple shape simulation data. Subsequently, the works published in [15, 16, 21, 23, 25, 29, 30] also demonstrated the potential of using neural networks to learn and generalize flow field information from training data, that has physical or geometric parameter diversity. However, most of these studies focused on capturing the geometric variation. Lee et al. [30] investigated the out-of-sample performance over Re number variation for a bluff body flow. Bhatnagar et al. [15] and Sekar et al. [16] studied the generalization across both physics and geometry. However, the studies were focused on a single flow regime based on Re. To make the models more informed about the underlying physics, Nabian et al. [17] introduced the physics informed regularization to the CNN model. The resulting network showed improved accuracy compared to the studies by Guo et al. [14] on the car flow prediction problem. However, this addition came at the cost of an increased computational burden and the requirement of handcrafting of features or equations. To the best of our knowledge, generalization of such methods across multiple flow regimes such as sub-sonic to transonic, incompressible to compressible, and low to high Re, at the same time having a considerable geometric variation, has not been investigated yet. Such generalization capacity, would be of immense utility in the aircraft design process discussed in the previous section.

1.2 Scientific Motivation, Knowledge Gap and Contribution

The methods described in the previous section have the potential to reduce the initial design and revision cycles for all processes that rely on CFD modeling for design. As an example, we consider the simulation and trajectory optimization of commercial and military aerial vehicles, which relies on CFD for the generation of large parametric databases. Dependant upon trajectory, such databases typically include integrated coefficients. However, the analysis of flow features and distributed field variables over is critical for thermal and acoustic loads. An example of what a comprehensive database would encompass at each point is provided by Orchard et al [31].

This CFD process, when performed on the entire geometry of vehicle, is computationally expensive and potentially takes weeks of simulation time [32] just for a single combination of flow conditions. For an entire database which covers multiple flow conditions, this process scales up to months. For such vehicles the physics involved gets increasingly complicated as the flight envelope expands. As Mach number moves from low subsonic, to transonic, to supersonic and hypersonic values, modeling must account for aerothermal, entropy and chemistry based effects. Such modeling is prohibitively resource hungry and precludes the development of comprehensive databases, thereby necessitating supplements in the form of expensive flight testing. The development of such databases using fully resolved and verified CFD data for canonical geometries is therefore, extremely appealing and is the primary motivation of this study. With an adequately sized, verified CFD database for flow over canonical shapes, a neural network-based model could be instrumental in mapping flow conditions and important geometric features onto complex vehicle geometries which can be reusable for multiple aircraft configurations. Such a similar concept of using data from previous design projects to aid the approximation of target domain has been previously looked at from an aircraft design perspective in the form of transfer learning [33]. With a potential to reduce the computational costs by at least an order of magnitude, these methods are worth exploring from design and synthesis perspective.

As demonstrated by the above studies, efforts in making the networks more generalizable, rely heavily on the learning algorithms to pick-up relevant features [14–16, 21, 29] from the input and then use them to make meaningful correlations. In comparison to methods aimed at improving the physics

based methods like CFD, the research published in exploring the generalization capability of Data-driven Deep Neural Network (DNN) models appears to be much less understood in literature. Due to the nascent nature of this application of learning methods, there is a gap in knowledge pertaining to their aptness in identifying and translating flow features and corresponding effects. The overarching goal of this study is to address some of the palpable and subtle concerns that arise when a generalization of predictive capability is sought across physics.

In order to meet this goal, our main objective is to establish the suitability of a convolutional encoder-decoder model (hereafter referred to as 'CNN based model' or just 'model'). While the general area of DNN has several advanced neural network architectures to offer, the CNN based model has been, so far, one of the popular models in the machine learning (ML) community for data-driven flow related predictions [14,15,17]. Especially this model is found to be useful for problems that encompass geometric variation, as well as changing flow parameters ***within same flow physics range***. The popularity is further evident as, with slight modifications to the above model, researchers [16, 18, 20–23] have used it for other flow prediction tasks, albeit a more geometrically varying data-set. However, the current literature does not focus on understanding of the failure modes, or comment on the reasons for loss in accuracy from the perspective of physics. Without understanding the impact that data driven modeling methods have on recognizing, retaining and reproducing flow features that begin to appear as Mach and Reynolds number change, it is difficult to conclusively argue their use in sophisticated aerothermal prediction processes. Hence, the current study is aimed to investigate the ability of a similar model to capture multiple physics at the same time. This includes flows over streamlined bodies (airfoils) that span over incompressible and compressible as well as laminar and turbulent flow ranges.

When trained using data that has diverse range of features, CNN encoder decoder structure is known to "blur" information in the process of capturing common underlying features in the data. Well documented cases of generalizability issues exist [34–36] in the context of computer vision and simple classification tasks. Extensive research has been conducted to investigate the effects of batch size [35, 37, 38], training methods, algorithms [34, 35] and theoretical reasons for cases in which deep, highly parameterized neural networks demonstrate some capacity to generalize [39]. However, there is a lack of consensus in terms of the traits of a good generalizable model [38, 40]. The encoder-decoder structure adopted in this study is similar to the structure seen in autoencoders which is used in many of these computer vision tasks and quite recently fluid tasks [14, 15, 17]. To the authors' knowledge, from a fluidic prediction perspective, where a generalizable model would mean generalizability across several flow applications/regimes and geometry, ***there is no standard benchmark set yet nor has this issue been investigated with quantitative detail***. Additionally, it is not clear how the deficiencies of the deep learning framework expressed in the literature cited above, would translate to the specific problem of flow field prediction. Although the current architecture used in this study relies on a deep encoder-decoder structure, the multiple outputs of the network (u, v and rho) replicate a multi-task learning problem and can advantage from the generalization capabilities of such models [41–43].

In this study, we observe similar concerns when trying to generalize across flow physics. Specifically, we demonstrate that when trained across varying geometries for a given flow regime (as defined by parameters such as the Mach and Reynolds numbers), the architecture generalizes adequately well (to an extent that they could be useful in CFD predictions in some future). However, the same level of generalizability is not achieved when training the network to learn the variations in the flow regimes. Furthermore, it is ***not obvious*** that the lack of adequacy in modeling flow regime transitions is exclusively because of an architecture inherent weaknesses. This points to a fundamental gap in literature that applies neural network models to flow problems. **To the best of our knowledge, our effort is the first to (a) expose this gap explicitly and quantitatively, (b) pin-point specific loopholes to avoid when utilizing such models for CFD-driven design tasks, and (c) demonstrate that regardless of their weaknesses, this class of architecture can still be utilized for design exploration if constrained through geometric variations.**

2 Materials and Methods

2.1 CFD Simulations and CFD Database Generation

In this work, the flow simulations are performed using a commercial StarCCM+ solver. The simulations are performed using the built-in steady-state compressible RANS, k-omega-SST model with wall functions for turbulence closure. The equations are solved using a coupled flow model to solve the mass, momentum and energy equations simultaneously using a pseudo-time marching scheme. The air is modelled as an ideal gas throughout the study. The simulation is carried out in a 2D setting with far-field boundary conditions at the boundary and no-slip boundary condition at the airfoil surface. Each simulation is run for 3000 iterations which either leads to a statistically steady state or the scaled residuals dropping by 3 orders of magnitude. It is key to note here that, because the ground truth for the CNN predictions is the CFD simulations and not the experimental data, the CNN model will not be able to outperform the CFD data. Moreover, the current study focuses on investigating the generalization capability only, and the above level of accuracy of CFD simulations is deemed to be sufficient to make such comparisons.

Fifty airfoil shapes which encompassed both cambered and uncambered airfoils were chosen for this study. This chosen database incorporates a variety of low-speed, high-speed, high-lift, etc. types of airfoils having important aerospace applications with characteristics similar to systems ranging from sail-planes to commercial airliners. In terms of variation in camber and thickness, the database encompasses a *maximum camber* variation from 0-8.4% and a maximum thickness variation of 6-24% (see Appendix for details of the precise airfoil geometries).

The simulations for these airfoils are performed at Reynolds numbers 100, 300, 500, 1000, 1500, 2000, 0.5e6, 1e6, 1.5e6, 2e6 and 3e6 at Mach numbers of 0.2, 0.3, 0.5 and 0.7. The low Reynolds numbers (Re $\leq$ 2000) cases for Mach 0.7 are removed from the simulation dataset as it leads to unphysical flow conditions due to high viscous heating. Each airfoil is tested for 6 angles of attack from $\alpha = 0$ to $\alpha = 10$. An unstructured mesh is generated for each airfoil geometry with 10 prism layers near the airfoil surface. Each geometry contains approximately 48,000 cells with a greater concentration of cells around the airfoil (Figure 1).

The combinations of above-specified parameters lead to the development of a dataset containing 11,400 simulations. This complete dataset is referred to as the master dataset hereafter. Such a database was designed in the context of design of aerospace type fluid system where interpolation techniques are used to obtain precise pinpoint solutions. At an appropriate level of density, it's desirable to obtain fairly accurate solutions using simple linear interpolation from such a database. Neural Network based architectures can differentiate themselves as it is possible to obtain a non-linear interpolation [44] which can offset the need to produce a dense dataset. This is also required as one needs to factor in the time take to train a model in addition to the time taken to generate the training data. The adoptability of learning methods in design domain will depend on how much information it can extract from a limited data.

The low Re cases, when simulated at an angle of attack, are characterized by a larger wake area behind the airfoil as compared to the high Re cases. The Mach 0.7 cases have regions in the flow field where the flow accelerates to supersonic flow and leads to a shock wave on the low-pressure side of the airfoil. The cases stated above are a few examples which elaborate the presence of varying flow features across the master dataset.

In addition to physically different flow regimes present in the master dataset, to test the geometric generalizability of the network, the dataset is also split into cambered and uncambered airfoils, as well as some corner and peripheral cases. The categorization is user-defined and is done specifically to identify the merit in such classification. This is also to investigate if a model trained on a smaller set of airfoils is sufficient to generalize across the entire distribution, and if not, the penalty incurred in reducing the samples is measured. A detailed description of the geometrically categorized classes is provided in the Results section.

2.2 Database for the CNN based model

For each study, an 80-20% split of each class is carried out for generating the training and test examples whenever the classes are segregated across physical parameters [45]. For geometric generalization studies, testing samples are chosen as few specific airfoils which are then excluded from the training data. More specific comments on the ways the training and test data is chosen for such studies is provided in the Results section where the classes and their results are discussed in succession to provide better coherence. For the application of the CNN based model, the data is sampled from the vicinity of the airfoil (Figure 1) in the form of a 256x256 Cartesian grid. The values of mean u and v components of velocity (components in the x and y direction) and the mean density are extracted from the simulation data. The extracted data is normalized across the entire dataset by performing min-max scaling.

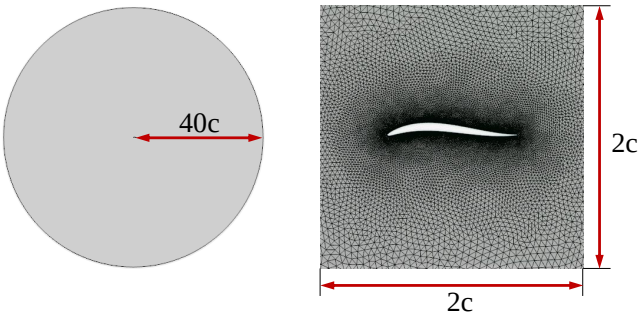

Fig. 1. Geometry. Left: The 2D control volume for the CFD simulations. Right: Zoomed view indicating the mesh and sampled area for training the model. (c = Chord length of the airfoils)

2.3 Convolutional encoder-decoder based model

Convolutional Neural Networks are a special type of Deep Neural Network specifically made for tasks classifying images. These types of networks, through convolution operation can compress the matrix form of data allowing data-compression. A Convolutional Autoencoder (CAE) is a special type of CNN where this compressed data is reconstructed to form the input image [46]. In this study, we use the encoder-decoder structure similar to the one found in the CAE. We pass the fluid flow parameter information to the compressed input data which passes through the decoder pipeline to produce the 2D flow field (velocity and density field). Thus the inputs of the model is the airfoil shape/geometry and the far-field data such as Mach number, Reynolds number and angle of attack and the output is the 2D flow field which compared to the flow field produced by the CFD. In the subsequent sections, we refer to this network structure as CNN based model for simplicity. At some occurrences, the word "CNN" is dropped when the context is clear. The details of the implemented CNN based model are now discussed.

The Signed Distance Function (SDF) array of the input airfoil geometry is encoded by the encoder which systematically picks up the relevant geometric features from the input geometry while downsizing the array. The final layer of the encoder is reshaped into a vector, to which a vector containing the information about the flow variables is concatenated. The input vector is split equally between the input parameters (3). The concatenated output is then fed to a fully connected layer. A shared decoder type of architecture as proposed by Bhatnagar et al. [15] is followed for decoding the encoded information and produce u and v velocity components and density field. The decoding layers consist of transposed con-

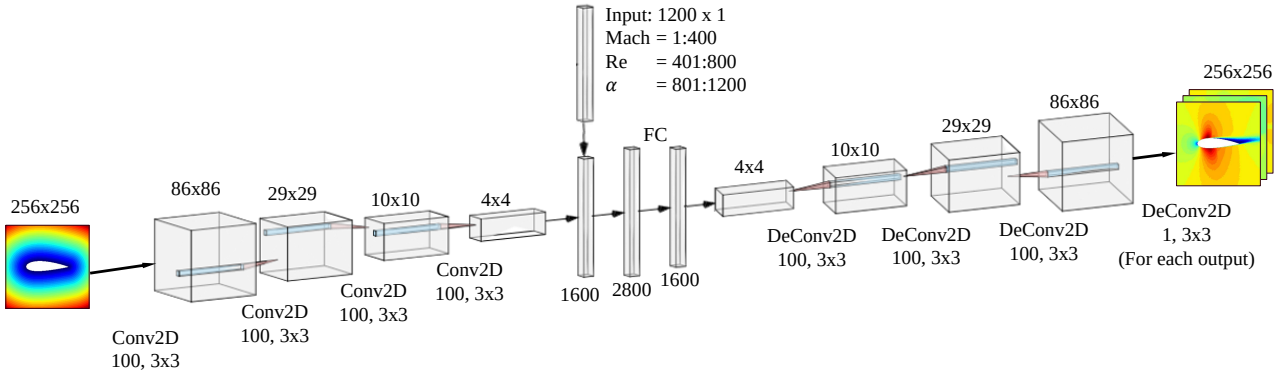

Fig. 2. Convolutional encoder-decoder based model architecture used in the current analysis

volutional layers which work the same as convolutional layers but in reverse.

An advanced Leaky ReLU function with $\alpha = 0.5$ is used for the activation in each layer. Each convolutional and transposed convolutional layer is followed by a batch normalization layer which normalizes the output of the previous activation layer [47]. This helps in speeding up the learning process as batch normalization reduces the covariate shift and reduces the effects of vanishing/exploding gradients. This also prevents the network from overfitting the training data which reduces the need for separate regularization.

2.4 Signed Distance Function

Signed Distance function (SDF) is used to provide the geometry information to the network. SDF is suitable for the CNN based framework, and its benefits over the regular binary representation are outlined in [14]. The SDF is normalized by dividing the maximum value across the entire data-set. The airfoil geometry is discretized using a Cartesian grid wherein all grid cells inside the airfoil are assigned -1 and those outside are assigned +1 . A Python module is then used which constructs the SDF for the geometry based on the fast-marching method [48]. The quality of the curve traced by the SDF is directly dependent on the resolution of the Cartesian grid and is one of the limiting factors for making an accurate representation of the geometry.

2.5 Training of Model and Architecture Variation

The different CNN architectures implemented in this study, along with their hyper parameters, are mentioned in Table 1. In order to account for the large complexity the model needs to handle in terms of the physics and geometry, a total of 100 filters were used in each layer in each of the network utilized. Architectures for CNN1, CNN2 and CNN3 represents the commonly used encoder decoder structure. Table 1 represents only the encoder side of the network. The decoder is the same as encoder but in the opposite direction as shown in Figure 2. For the network with skip connections, the input from the corresponding layer in the encoder is also fed to the decoder along with the input coming from the previous decoding layer. The variations in architecture and hyperparameter settings also resulted in the variation of latent space vector.

CNN2 exhibited lower training and validation loss in comparison to CNN1 and CNN3. Although CNN4 provided marginally better training and validation losses, the skip connections led to an additional computational cost. Decreasing the latent vector size seemed to decrease the accuracy as well as the computational time due to reduction in the size of fully connected network (CNN1). However, a more thorough study is necessary into this in order to conclusively ascertain the effect of latent vector size. Based on these observations, CNN2 seemed to strike the right balance in terms of computational cost and accuracy and was utilized as the primary model for all the further studies.

The network (CNN2) is trained using a batch size of 32 examples for 150 epochs for each case as the loss curve flattened beyond 150 epochs (Figure 3). The optimization is carried out using the RMSprop optimizer. The learning rate is initiated at 0.001 and is decayed by a factor of 0.1 when the learning reached a plateau, i.e. the learning stagnated for the last 5 iterations. The batch normalization layer used, makes use of the mean and variance for the current batch while computing the training error. However, it uses the running mean and variance for evaluating the validation dataset. Thus the validation error is lower than training error as the mean and variance are different for each batch which is not representative of the entire dataset. The removal of batch normalization layers causes the two errors to match more closely (Figure 4).

3 Results and Discussion

3.1 Metric for Comparison

The Mean Absolute Percentage Error (MAPE) metric was chosen to compare the different studies. This metric allows comparison of quantities in terms of relative error. MAPE is made to compute the relative error between the ground truth CFD results and the flow field generated by the model.

The proposed CNN based model is evaluated based on multiple datasets that are sub-sampled from the master dataset. The MAPE evaluated on the test dataset for each model

Table 1. CNN architectures with their hyper parameters (Losses are reported for Mach 0.7 predictions)

Layer type	CNN1	CNN2	CNN3	CNN4 (Skip Conn.)
Input	256x256x3	256x256x3	256x256x3	256x256x3
1st Convolution	3x3, 100	3x3, 100	5x5, 100	3x3, 100
2nd Convolution	3x3, 100	3x3, 100	5x5, 100	3x3, 100
3rd Convolution	3x3, 100	3x3, 100	5x5, 100	3x3, 100
4th Convolution	3x3, 100	3x3, 100	...	3x3, 100
5th Convolution	2x2, 100	...	...	...
Latent Vector	1x400	1x1600	1x1600	1x1600
Input Vector	1x300	1x1200	1x1200	1x1200
Training time	5.56 s/epoch	17.72 s/epoch	4.94 s/epoch	18.36 s/epoch
L2 loss - Training	7.59e-4	6.71e-4	8.31e-4	6.64e-4
L2 loss - Validation	6.86e-4	6.00e-4	8.01e-4	6.02e-4

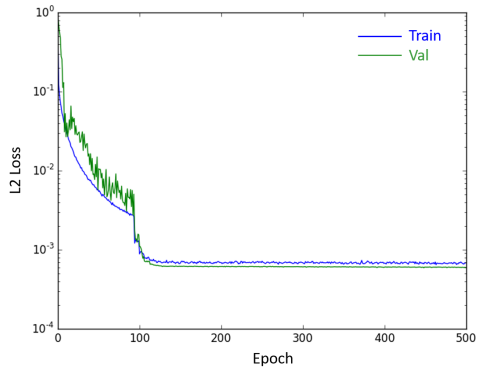

Fig. 3. Plot showing training and validation losses for a model trained on Mach 0.7 data

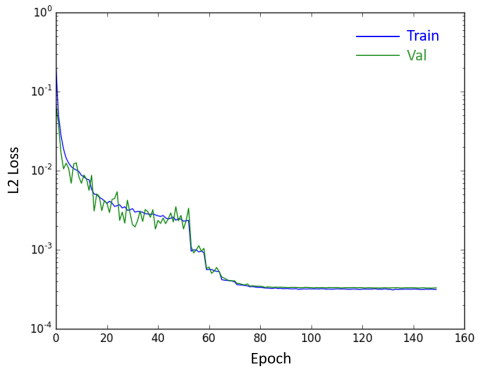

Fig. 4. Plot showing training and validation losses for a model without batch normalization

trained using different sub-samples is recorded for relative comparisons. Whenever an out-of-sample reference in this study is made, it refers to a sample that is not only absent in the training data, but is also significantly different from training data in terms of the geometry/representative flow field variable (based on the context). In the subsequent sections, the observations based on such out-of-sample performance of the models trained on sub-samples of data is discussed in detail.

3.2 Prediction performance across compressible and incompressible flow regimes

To test the model's ability to fit data which is categorized based on the compressible/incompressible flow regime, the master dataset is sub-sampled based on the Mach number. Individual datasets containing simulations for a single Mach number are generated which is further split into 80% for training, and 20% for testing the trained model. Figure 5 summarize the comparison of the predictive performance of

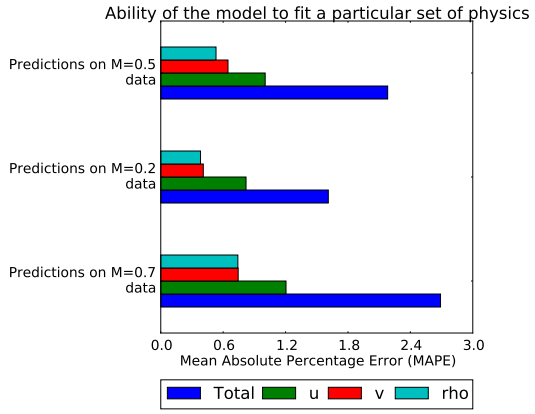

Fig. 5. Performance of the model at various Mach numbers

the CNN model when trained on these multiple datasets.

Overall, the model delivers an average of 2.16 MAPE on all Mach numbers (the largest being 2.69). A more detailed

comparison between the 3 studies reveals that the accuracy of predictions declines as the Mach number of the training examples increases. The prediction indicates that the model is unable to track the sharp changes in gradients, which are prominent in the Mach 0.7 case due to stronger separation between low and high-velocity regions, as well as the shock wave located on the low-pressure side of the airfoil. The exact position and the strength of the shock wave are different for each airfoil. Moreover, the size of the shock wave is relatively small when compared to the larger flow field features like the wake, stagnation and low and high-pressure regions and hence the model does not accurately capture weak shocks from the flow field.

The gradient information across the shock wave is converted to a localized gradient which is not always a contrasting factor in the entire flow field (Figure 6). The weak shock wave which appears as a sharp discontinuity in the CFD data appears to be smeared in the CNN predictions (Figure 6). This suggests that, even when a true matrix of primitive variables (u, v and rho quantities form CFD) is utilized, the CNN based model is unable to preserve these sharp gradients. This is likely due to the relative size of the localized gradient as well as the filter size and strides taken during the convolution operation. The fact that the size of such disturbance is quite small for low transonic Mach numbers, makes it difficult for the CNN based framework to capture the weak shocks accurately. Accurate tracking of weak and strong characteristics of this nature is important for achieving accurate predictions in the transonic domain. It would however, require additional effort during the training process, to improve the tracking of such small disturbances in the context of gradients that exist in the larger flow field.

3.3 Generalization across physical parameter variation

3.3.1 Mach number variation

To study the generalization across the Mach numbers, multiple datasets are created. Each of these datasets contained samples which have the same Mach number as that of the test cases. However, no sample is exactly the same in the training and the test dataset. The samples in test and training dataset varied in other parameters like the Reynolds number, airfoil shape and the angle of attack. The datasets were progressively added with training samples of additional Mach numbers apart from the test Mach number. As an example, for testing the predictions on Mach 0.7 samples (Figure 7), first, a model is trained using only the Mach 0.7 training data. After this, Mach number variation is added (Mach 0.2 and 0.3 examples in addition to Mach 0.7 ones, etc.) to analyze the ability of the model to generalize across Mach number variation. The Δ MAPE was less than 0.5 when generalization is tested on Mach 0.7 and Mach 0.2 test samples (Figure 7 and 8). The test data is kept consistent when evaluating the multiple models trained in this section. Each model encompasses a greater Mach number variation (Figure 7 and 8) and they are tested on 20% of the chosen Mach number's total samples (20% of total Mach 0.7 samples for models in Figure 7 and 20% of total Mach 0.2 samples for models in Figure 8). The out-of-sample performance for the Mach number is not evaluated, meaning the training data always consisted the samples which had Mach number representative of the test data.

It is observed that the flow field is quite similar up to a Mach number of 0.5. At Mach 0.7, the presence of the shock wave becomes prominent. Not only does the model fail at accurately reproducing the shock wave, the addition of Mach 0.7 data to the training data also degrades the performance for predictions on Mach 0.2 test data. This is because the model's parameters are now trained to fit the flow features of two different types of flow regimes. For Mach ≤ 0.3, the density field is constant while for Mach 0.5 and 0.7 cases, significant variations are observed within the density field due to compressible flow. This implies that the CNN based framework does not capture the variations in density at higher Mach numbers (Mach > 0.3) as accurately as it does for the lower Mach numbers (Mach ≤ 0.3).

Moreover, the addition of more Mach variation outliers to the training data affects the prediction accuracy for Mach 0.7 more severely than it does for the Mach 0.2 predictions. This behavior can be attributed to the master dataset being more biased towards low Mach cases which are devoid of a shock wave in their flow field. The aggregate MAPE based loss in accuracy incurred by using a generalized training model is on an average 23.95% as compared to the model trained on the specific class of Mach number. Closer inspection of the flow field for the various Mach numbers shows that except for the density field, and the presence of shock wave for Mach 0.7 cases, the larger flow field features like the wake region, stagnation and low/high-pressure regions are similar from a visual standpoint. Thus, the model appropriately captures these larger flow field features which have relatively minor variation for Mach number up to 0.5. To the model, the variation in physics that appears in the form of shocks and other transonic flow features is indistinguishable from other visually based intensity and gradient changes that a CNN type architecture is designed to extract. Therefore, for generalization to be successful in this context, an advanced flow feature extraction methodology is required. Such a methodology should distinguish between the above mentioned (shock and wake) flow features. This result also implies that intelligent sub-sampling of a large database based on physics will lead to a noteworthy improvement in prediction accuracy. Alternatively, generation of a training data set that is guided and reduced by knowledge of such features, will require less resources while being more accurate.

3.3.2 Reynolds number variation

To study the generalization across the Reynolds numbers, multiple datasets are generated, with either low (Re ≤ 2000) or high (Re > 2000) Reynolds number cases only. A generalized dataset capturing the entire Reynolds number range is also generated to make the generalization comparisons. These models are then tested for predictions on high Re cases. The dataset is split into two low and high Re simulations. For evaluation, two high Re sub-samples at different Mach numbers are chosen. The test data is kept the same for the

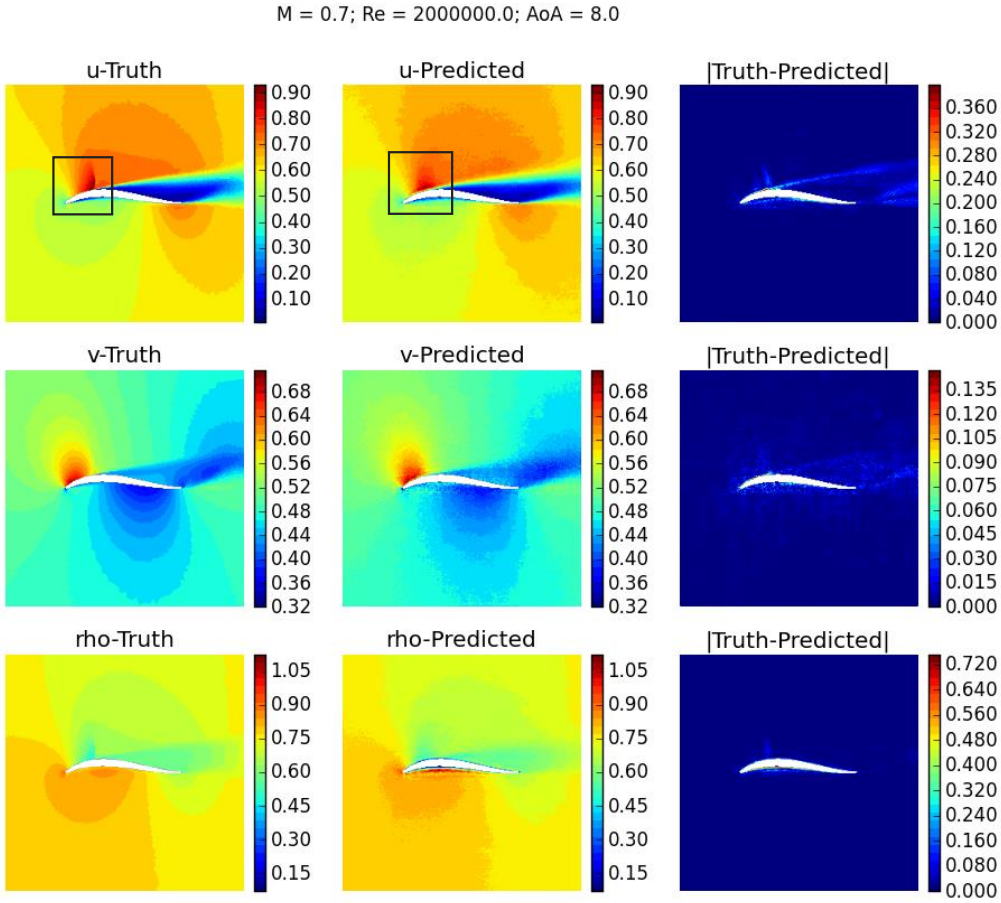

Fig. 6. Flow field predictions for Mach 0.7 case (scaled). Left: Ground truth from the CFD simulation. Center: Predicted flow field from the model. Right: Absolute error between the ground truth and the CFD simulation. The black boxes highlight the weak shock present in the flow field (left) and the failure of the CNN based model to capture it appropriately (center).

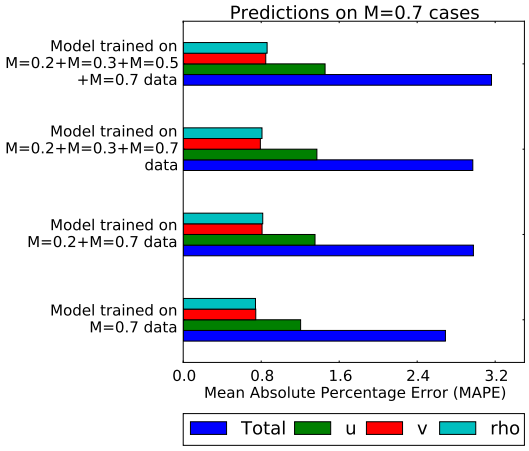

Fig. 7. Performance of the model for predictions on Mach 0.7 case

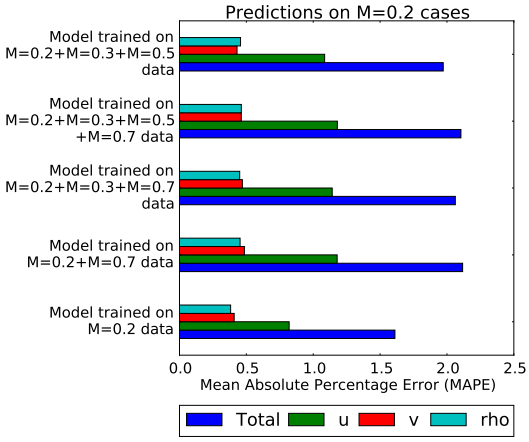

Fig. 8. Performance of the model for predictions on Mach 0.2 case

respective Mach numbers, meaning, for testing the predictive performance on high Re cases at Mach 0.2, models are trained only using Mach 0.2 data. The variation is therefore limited to Reynolds number, angle of attack and airfoil geometry. The training model is varied such that only high Reynolds number simulations are used for one while for the other, all the Reynolds numbers pertaining to that specific Mach number are included in the training data.

The penalty incurred in MAPE by generalizing across the Reynolds numbers is found to be 25.56%. This is higher as compared to the previous case where the generalization was focused to capture the Mach number variation (Figure

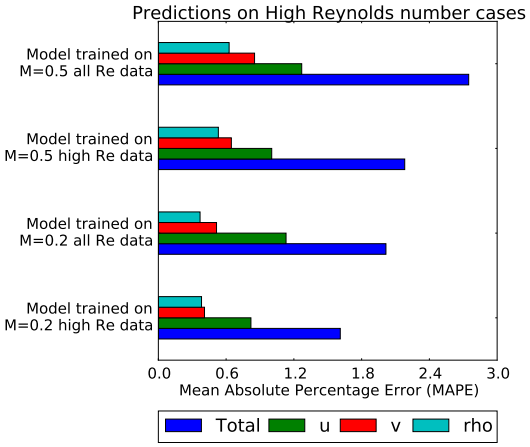

Fig. 9. Performance of the model for predictions at various Reynolds numbers

9). This behaviour can be attributed to significantly different flow features that are dominant in each of these flow regimes (Figure 10). As the flow field is dominant in one direction, the wake sizes can be seen explicitly with just the representation of the u-component. Hence, the v-component and the density fields are not represented in Figure 10 and 11 while making the comparisons based on wake sizes.

The low Reynolds number simulations are characterized by a larger wake with a smoother gradient between the low and high values. On the other hand, the high Reynolds number simulations are characterized by relatively smaller wake and sharper gradient in values (Figure 10). As the flow features are significantly large (unlike the shock wave) and also widely different from a visual standpoint, training a generalized model to capture these flow features is found to be difficult. The errors in predictions (MAPE) increases by approximately 25% for each Mach number whenever a single model was fit to generalize the Reynolds number variation. The 2D matrix representation of the data of the standard primitive variables is not comprehensive in capturing these strong variations. A higher level representation of the data that accounts for more physics, could perhaps be used to alleviate this issue. It is observed that the maximum error in the flow field prediction is for the u-component of the velocity. Moreover, the regions of high error in the u-component match closely with the areas of high vorticity (Figure 11). This suggests that the errors are high in the regions where vorticity diffusion is strong. Flow regions in which there is high shear or large pressure gradient are typically responsible for vorticity generation. This indicates that the popular architecture fails to capture such effects in the absence of an additional constraints that are rooted in physics. For 2D cases, the vorticity is a function of velocity components, and this information is available for to the network in making predictions. Failure to preserve such effects highlights the inability of this data-driven method to relate cause (shear within the boundary layer) with effect (gradient change in velocity field). This further suggests that the application of such methods would require additional fine-tuning based on domain specific knowledge, as one cannot completely rely on the architecture to preserve such effects.

A sub-study (using 1800 samples), analyzing the influence of explicitly adding vorticity to the loss was conducted to determine if it was possible to enhance the ability of the architecture to better develop causal correlations. This additional term in the loss function did not result in a significant gain in accuracy (see Appendix for details on this sub-study). This result does not come as a complete surprise since vorticity is a derived quantity dependent upon the velocity gradient information, which is inherently present in the analysis conducted based on velocity and density alone.

3.4 Generalization across geometric variation

To study the effect of generalization across geometry, the prediction performance of the CNN based model is evaluated for un-cambered and cambered airfoils. For this study, the test data is an un-cambered airfoil selected randomly from the 11 un-cambered airfoils present in the master dataset. Multiple models are then trained using multiple different sets of airfoils (Figure 12). The goal of this study is to investigate the out-of-sample performance of the models trained on the cambered airfoils as tested on an un-cambered airfoil.

To have sufficient number of training samples in each sub-dataset, all the physical parameter variation corresponding to a particular airfoil is included. This implies that the model is already generalizing across the entire physical variation, which is the reason for relatively larger values of MAPE as compared to the ones observed in the physics based studies. Nevertheless, as the comparisons are across geometry and the fact that the physical-based variation is consistent in all the data samples, it is reasonable to imply, that this limitation does not affect the overall trend of generalization.

It is observed that the model trained on 10 un-cambered airfoils provides better prediction accuracy compared to the model trained on just 10 cambered airfoils (Figure 12). A 26.36% difference in the predictions made by these two models indicates that the model trained on just cambered airfoil data is insufficient for making predictions on un-cambered airfoils. However, it is interesting to note that by just the addition of 10 more cambered airfoils, the predictive performance is at par (MAPE difference increase is just 1.03%) with the model trained on the 10 un-cambered airfoils for the un-cambered airfoil flow prediction. This out-of-sample performance indicates that the model is capable of understanding the underlying geometric variation beyond the proposed categorization based on the camber. This further highlights the performance of the CNN based model on making predictions on completely unseen geometry.

In the subsequent studies, the performance further improves when more training data is added (Figure 12). Interestingly, the addition of such a large sample of cambered airfoils does not let the model over-fit the cambered airfoils simulations. Instead, it improves the performance leading to a reduction in MAPE by 12.40% compared to the model trained on un-cambered airfoils alone. This fact provides some insight regarding the potential mechanism that ascertains how the network extracts geometric features from input shapes and

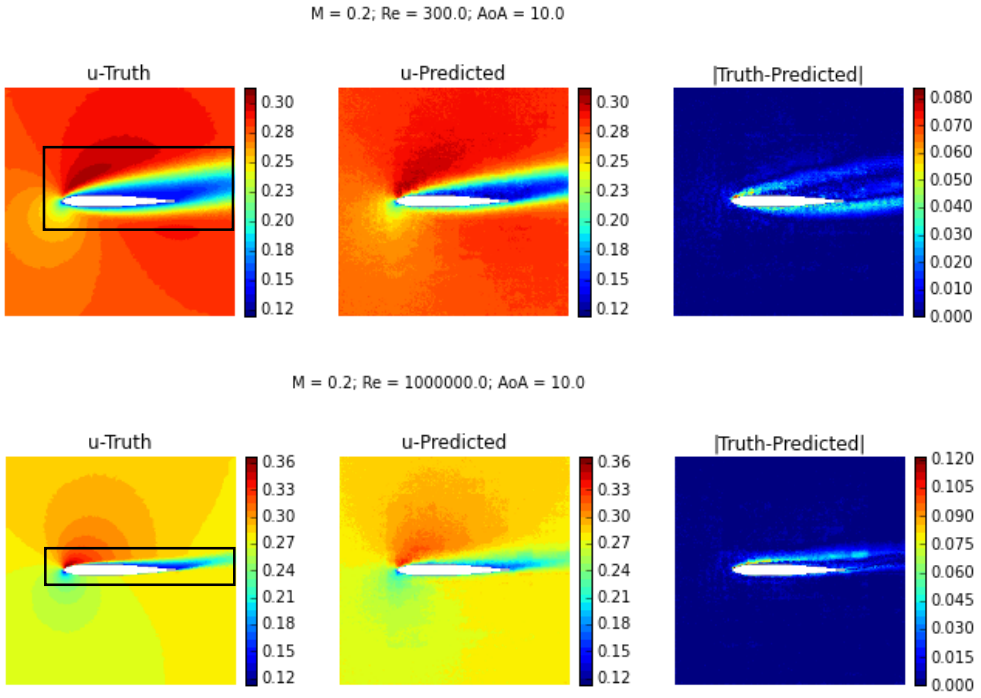

Fig. 10. Flow field comparison between low and high Reynolds number simulations (Scaled). Top: Predictions for low Reynolds number case. Bottom: Predictions for high Reynolds number case. Left: Ground truth from the CFD simulation. Center: Predicted flow field from the model. Right: Absolute error between the ground truth and the CFD simulation. The black boxes highlight the wake regions in the flow field

ties them to solution physics. Instead of directly associating the input with the output flow field, the auto-encoder type architecture allows the input geometry to be reduced into few meaningful geometric features, which are then mapped onto flow field data along with the input from the flow field boundary conditions. Nonetheless, the demonstration of generalization capability and reasonable out-of-sample performance implies that the extracted geometric features are common to both cambered and un-cambered airfoils. Some examples of such features could include the thickness and curvature which are common for both cambered and un-cambered airfoils.

3.5 Dependence of geometric generalization performance on dataset size

The Section 3.4 depicts how the chosen CNN based model can generalize across the geometry for making predictions across completely unseen airfoil geometry. The predictions on the un-cambered airfoils presented in the Section 3.4 also indicate that the encoder-decoder type of architecture allows systematic extraction of geometric features from the input shape. This opens up avenues to reduce the geometrically varying samples provided the chosen geometric features are representative of the overall variation/distribution.

To study this, various cases are designed containing samples that capture various geometric details of the airfoil data. The distribution of the airfoils chosen for training the various models used in this section is summarized in Figure 13. The test data is an airfoil that represents the mean of all the airfoils in the master dataset when plotted based on their maximum thickness and curvature values on a thickness vs. camber plot. Case 1 represents the airfoils that have maximum/minimum camber or thickness. Case 2 is an extension of Case 1 with 4 additional airfoils along the periphery of the polygon formed by the corner airfoils on the Thickness-Camber plot. Case 3 represents a biased dataset i.e. the chosen airfoils have either same camber or thickness as the test airfoil. Case 4 represents a case of a random sampling of 8 airfoils. As the sampling method is random, there is a possibility of the sampled airfoils being well distributed across the test airfoil or being concentrated in a particular region. In fact, Case 4 represents such an instance of concentrated distribution of the airfoils, situated only in low thickness area. Figure 13 which excludes Case 5, represents random distribution of airfoils with uniform distribution across the test airfoil. These strategically chosen cases allow study into the possibility of sample size reduction as stated above. Case 1 and case 2 are of special importance. It is believed that for the current characterization of the airfoils (using thickness and camber), these cases should ideally be sufficient to capture the encompassing geometric variation in the master dataset. If a model trained on airfoils mentioned in such cases is able to produce reasonable prediction accuracy, the number of geometrically varying training samples can be significantly reduced. The performance of the model based on the above cases is now discussed in detail.

A model is trained on each one of cases mentioned above and their predictive performance is evaluated against the test average airfoil shown in Figure 13. The model trained the cor-

Table 2. Mean Absolute Percentage Error (MAPE) values for the CNN predictions when trained using different choices of airfoils

Case	Description	MAPE (%)				Total MAPE / # of airfoils
		u	v	rho	Total	
Case 1	Model trained on 4 corner airfoils	15.768	11.113	17.098	43.980	11.00
Case 2	Model trained on 8 peripheral airfoils	3.577	2.049	1.324	6.950	0.87
Case 3	Model trained on 8 airfoils (same camber or thickness)	3.312	1.775	0.961	6.048	0.76
Case 4	Model trained on 8 random airfoils	4.409	2.776	1.731	8.915	1.11
Case 5	Model trained on 16 random airfoils	1.438	0.721	0.761	2.920	0.18

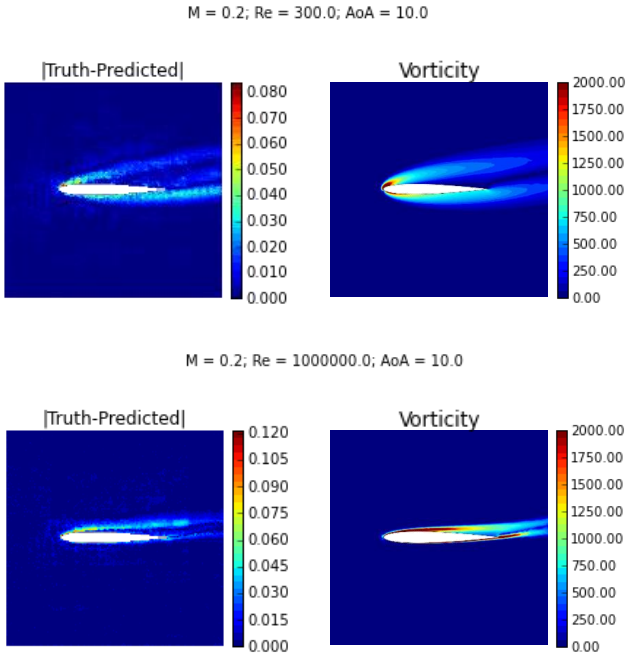

Fig. 11. Comparison between the u-velocity absolute error (scaled) and Vorticity field from CFD. Top: Data for low Reynolds number case. Bottom: Data for high Reynolds number case

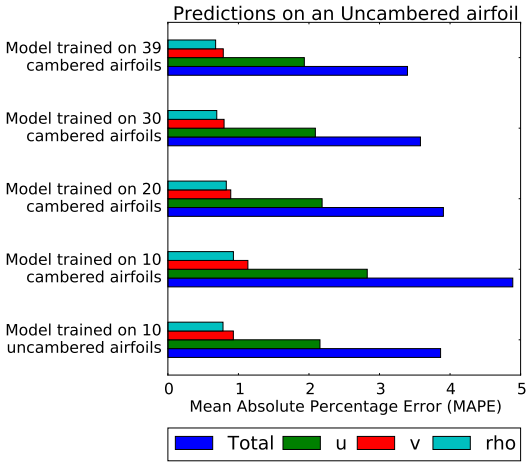

Fig. 12. Performance of the model for predictions on an uncambered airfoil

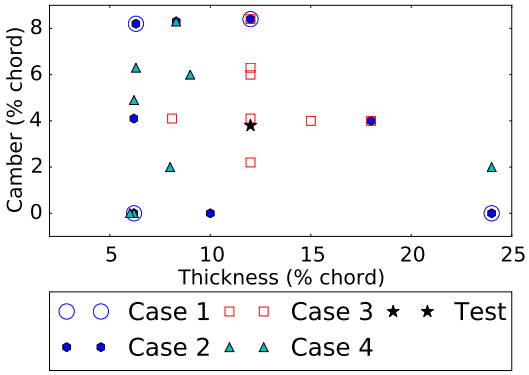

Fig. 13. Airfoil distribution for training the different models

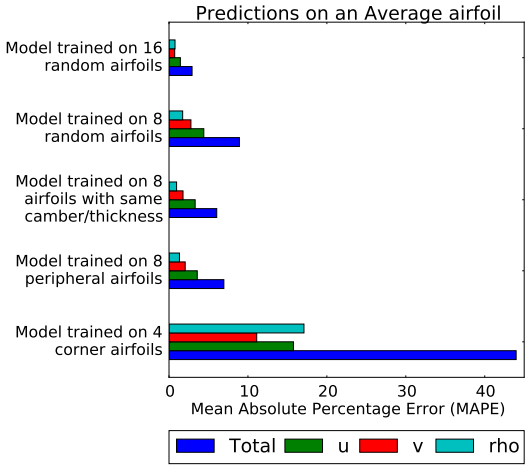

Fig. 14. Performance of the CNN model when trained using different choices of airfoils

ner airfoils has the maximum MAPE (43.98%) and is found to perform poorly as compared to the other cases where the MAPE is less than 10% (Table 2, Figure 14). Moreover, the model trained on 16 airfoils performs better compared to other studies. This indicates that the accuracy is indeed directly related to the number of training samples, when making predictions across the geometric variation. The model trained on the 8 peripheral airfoils (Case 2) performs better than even a biased model (Case 3) and although the error is almost

Table 3. Conservation of mass comparison based on imbalance as measured for CFD vs CNN prediction

	Mass imbalance (kg/s/m)
True	0.164
Predicted	2.981

double compared to a model trained on 16 airfoils (Case 5), (MAPE for Case 5=2.92% and MAPE for Case 2=6.95%) the predictions are within reasonable limits of accuracy given that each airfoil contains only 228 flow cases. All the models in this section are trained with a relatively smaller training dataset compared to the models trained in the physics-based classification and this is the reason behind the MAPE comparisons being more conservative compared to the ones made in physics-based classifications.

The best [MAPE/# of airfoils] ratio is obtained for a model trained on a larger number of samples for obvious reasons. However, this ratio for Case 2 ([MAPE/# of airfoils] = 0.87) is very close to that of Case 3 ([MAPE/# of airfoils] = 0.76), which represents the best distribution of the 8 airfoils given the test airfoil. A comparable [MAPE/# of airfoils] ratio of Case 2 and Case 3, shows the capability of Case 2 in capturing the underlying geometric variation reasonably. Thus, Case 2 is quite beneficial in terms of reducing the computational costs while maintaining reasonable levels of accuracy. A model trained on fewer data points implies not only cost savings in training the model, but also cost savings in generating the required CFD cases. This behavior of the CNN based model to pick up the relevant geometric features from minimal data, unlike flow field features like shocks, can be leveraged in design studies as a considerable amount of time is required to generate each sample in the CFD studies. If an appreciable level of accuracy can be achieved by a lower yet strategically planned sample, it would benefit the initial design-space exploration and can bring down the overall design cycle time.

Sections 3.4 and 3.5, which focused on studying the ability of the model to capture the geometric variation, suggest that the model is capable of generalizing across the variation of geometry without overfitting to a particular class of airfoils. Lastly, predictive performance of the model almost always improves with the addition of more training data.

3.6 Mass conservation results

An application of such a data-driven method to physical simulations like these would also require an appreciation of the governing laws and invariances that are particular to the studied physics [1]. To evaluate this, the model's performance is measured to check the compliance of the mass conservation law. For computing the mass imbalance, the variables are scaled back to original values. All the comparisons are made with respect to the mass imbalance computed on the CFD simulations that form the ground truth for this study. The results of mass conservation computed across the 2c x 2c square domain defined around the airfoil geometry in the flow field are shown in Table 3. The current CNN based model, without explicit definition of conservation of mass as a loss function, performs poorly for the most part, in obeying the conservation law with the predictions being off by up to 13x from the true values. There are techniques as suggested by [17] to incorporate these laws into the optimization framework by means of loss function. However, such addition comes at an additional computational cost even when it is just in training. Potentially, this can be tackled either by generating a denser dataset or adding extra constraints on a sparse dataset. Thus, through this study, the assessment of the network's ability to pick up such physical effects, without explicit specification of governing physics during the training process is made. The wide difference in the predicted and the true results demonstrates the drawbacks of using such CNN based techniques as they are. This necessitates either the incorporation of such laws into the CNN framework or generation of a denser dataset, both resulting in an additional cost.

4 Conclusions

In this research, we implemented a popular CNN based framework in fluid flow field prediction and extended it to various sets of physical parameters and geometric variation. To the best of our knowledge, such a comprehensive study is the first one of its kind. With incorporation of multiple flow physics effects into the dataset, this study tackled the suitability of this popular method in the context of generalizability as opposed to development of a finely tuned model for a specific application around a fixed set of physics.

A model trained on the entire master dataset is tested at various categorically sampled test datasets. The Mean Absolute Percentage error is found to be less than 5% for each test dataset when compared to CFD. Thus it can be concluded that such a network architecture adequately captures the larger flow field patterns like wake flows or the regions of stagnation. However, as it stands, it fails to capture sharp gradient changes like the ones at wake boundaries or weak shock waves occurring on the suction side of an airfoil in transonic cases. Additionally, it struggles with the mass conservation in many cases. Generalization loss associated with the available Mach number range was approximately 24%, when compared to model that was trained on Mach specific data. While this is better than certain analytical predictive tools there is a noteworthy margin when compared to CFD methods, which are already approximate in nature. Additionally, the current dataset was slightly biased towards Mach numbers in the subsonic region with only the Mach 0.7 case having some regions of supersonic flow. Thus, comprehensive positive claims about the overall generalizability of the model for diverse Mach number based physics predictions, would require further analysis. Similarly, for the generalization across the Reynolds number range, a 25% loss in predictive performance (compared to model trained on specific Re data) was observed across all Mach number variations when tested using a generalized model. This behavior is attributed to the widely varying flow features across the Re range (e.g. wakes,

separation etc.). From these studies, it can be concluded that overall the CNN based model suffers reduced fidelity when generalizing across sub-sets of physics variation.

The out-of-sample performance of these models in generalizing over various flow regimes is extremely poor (ex. the MAPE for a model trained on low Reynolds number data used for testing on high Reynolds number cases was found to be 14.39% when compared to CFD). This is due to the widely varying patterns in each of these regimes.

It can also be concluded that accuracy associated with learning methods in the fluids domain depends on the data sub-sampling technique used for a particular prediction task. In other words, generalizability comes at an additional cost. This cost can be in terms of accuracy or computational expense, if one decides to add extra features to support the multiple physics. In general, a degradation of performance is observed when a network trained on the full dataset is used to make highly specific predictions. This indicates that there exist far too many, subtle or flow regime specific flow features for the network to capture. This also suggests that the popular CNN architecture used for this study, does not reduce the dimensionality of physical parameter space, in a manner similar to that of geometry. It can therefore be concluded, that one has to intelligently produce samples in each flow regime to allow the network to train equally on each physical parameter.

Most importantly however, we come back to the relative success of CNN encoder-decoder structure in flow prediction for geometric variations. In this study, we demonstrate avenues by which this architecture can be fruitfully utilized regardless of the architecture's deficiencies. The CNN encoder-decoder architecture does indeed exhibit a potential to generalize better across geometric variation, albeit with reasonably invariant flow physics. The results shown in Figure 12, support this conclusion. In this case a prediction was sought across a uncambered airfoil. When the prediction was made using using flow data from 10 cambered airfoils the results contained higher error than for the case when the prediction was made using 10 uncambered airfoils. However when the same prediction was made using 39 cambered airfoils, the results were more accurate than the 10 uncambered airfoil case. This knowledge facilitates the possibility of application of this architecture, and other similar architectures, as a design tool. As previously mentioned, for an aerial system flying at any nominal trajectory point, it is not uncommon for different parts of the vehicle to be experiencing different sets of flow physics (e.g. stagnation point on the nose vs. a location downstream of some protrusion). The generalization concern across physics for such a circumstance, can be alleviated by utilizing a pre-classifier at a higher level, that identifies the ***appropriate local physics*** based on a nearest neighbor match that exists in a pre-generated sparse CFD database for the aerial vehicle. Once the physics is identified, a model that is trained over ***only such physics*** using a much denser dataset of diverse random canonical shapes can be used to make an accurate prediction for the locality of interest. With this approach to design we are not proposing that the learning architecture be used as a replacement for CFD. Instead it supplements the established design process by drastically reducing the size of the initial CFD database that is required. We find that this insight, which can be quite useful for real-world design applications in the fluidic domain, is not readily and explicitly available in current literature.

While we did observe a 12.4% improvement when prediction was made with a generalized training set, it possible that this is specific for geometries similar to the subset of airfoils tested. For more sophisticated or composite geometries further analysis will be undertaken as part of future work. Notwithstanding the limitations, there is significant utility in the selected airfoil dataset, as they represent variations encompassing several important aerospace systems.

The results of the current analysis indicate that the model can generalize the geometric space of chosen airfoils reasonably well. Moreover, it is shown that a reduced yet intelligent sampling of the training dataset can lead to good levels of accuracy at a fraction of the computational cost compared to the models trained on a larger datasets.

Finally, the geometry versus physics generalization study performed herein, provides a road-map of the strengths and weaknesses of prediction, using generalized data-sets, in fluidic domains that contain variation in physics of the flow regime and object geometry.

Acknowledgements

The authors would like to thank the Texas A&M University Engineering Experiment Station for providing funds that were used to conduct part of this research. The authors would also like to acknowledge the support of Peng Jiang, Kartik Prakash, Mohammad Naveed Gul, Shah Akib Sarwar, Madhu Areti and Briana Holton at the Texas A&M University, for their support with setting up CNN architecture and editing. Finally, the authors would like to thank Dr. Nima Kalantari at Texas A&M University for his valuable insight in developing better representations of the training data.

References

[1] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., 2020. "Machine Learning for Fluid Mechanics". *Annual Review of Fluid Mechanics,* **52**(1), pp. 1–31.

[2] Yumer, M. E., Asente, P., Mech, R., and Kara, L. B., 2015. "Procedural modeling using autoencoder networks". In Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology, UIST '15, ACM, pp. 109–118.

[3] Chen, W., Fuge, M., and Chazan, N., 2017. "Design manifolds capture the intrinsic complexity and dimension of design spaces". *Journal of Mechanical Design,* **139**(5), pp. 051102–051102–10.

[4] Zhang, W., Yang, Z., Jiang, H., Nigam, S., Yamakawa, S., Furuhata, T., Shimada, K., and Kara, L. B., 2019. "3d shape synthesis for conceptual design and optimization using variational autoencoders". In ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering and Conference, American Society of Mechanical Engineers.

[5] Tracey, B., Duraisamy, K., and Alonso, J. J., 2015. "A machine learning strategy to assist turbulence model development". *53rd AIAA Aerospace Sciences Meeting*(January).

[6] Ling, J., Kurzawski, A., and Templeton, J., 2016. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". *Journal of Fluid Mechanics,* **807**, pp. 155–166.

[7] Singh, A. P., Medida, S., and Duraisamy, K., 2017. "Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils". *AIAA Journal,* **55**(7), pp. 2215–2227.

[8] Bar-Sinai, Y., Hoyer, S., Hickey, J., and Brenner, M. P., 2019. "Learning data-driven discretizations for partial differential equations". *Proceedings of the National Academy of Sciences,* **116**(31), pp. 15344–15349.

[9] Loiseau, J. C., and Brunton, S. L., 2018. "Constrained sparse Galerkin regression". *Journal of Fluid Mechanics,* **838**, pp. 42–67.

[10] Raissi, M., and Karniadakis, G. E., 2018. "Hidden physics models: Machine learning of nonlinear partial differential equations". *Journal of Computational Physics,* **357**, pp. 125–141.

[11] Raissi, M., Perdikaris, P., and Karniadakis, G. E., 2019. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics,* **378**, pp. 686–707.

[12] Lecun, Y., Bengio, Y., and Hinton, G., 2015. "Deep learning". *Nature,* **521**(7553), pp. 436–444.

[13] Kutz, J. N., 2017. "Deep learning in fluid dynamics". *Journal of Fluid Mechanics,* **814**, pp. 1–4.

[14] Guo, X., Li, W., and Iorio, F., 2016. "Convolutional neural networks for steady flow approximation". *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* **13-17-August-2016**, pp. 481–490.

[15] Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., and Kaushik, S., 2019. "Prediction of aerodynamic flow fields using convolutional neural networks". *Computational Mechanics,* **64**(2), pp. 525–545.

[16] Sekar, V., Jiang, Q., Shu, C., and Khoo, B. C., 2019. "Fast flow field prediction over airfoils using deep learning approach". *Physics of Fluids,* **31**(5).

[17] Nabian, M. A., and Meidani, H., 2018. "Physics-Informed Regularization of Deep Neural Networks". pp. 1–19.

[18] Liu, Y., Lu, Y., Wang, Y., Sun, D., Deng, L., Wang, F., and Lei, Y., 2019. "A CNN-based shock detection method in flow visualization". *Computers and Fluids,* **184**, pp. 1–9.

[19] Chen, Q., Zeng, Z., and Lian, L., 2018. "Accelerating underwater vehicle hydrodynamic analysis using convolutional networks". *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans, OCEANS - Kobe 2018*, pp. 1–5.

[20] Hennigh, O., 2017. "Lat-Net: Compressing Lattice Boltzmann Flow Simulations using Deep Neural Networks".

[21] Viquerat, J., and Hachem, E., 2019. "A supervised neural network for drag prediction of arbitrary 2D shapes in low Reynolds number flows".

[22] Zhang, Y., Sung, W. J., and Mavris, D., 2018. "Application of convolutional neural network to predict airfoil lift coefficient". *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018*(210049), pp. 1–9.

[23] Miyanawala, T. P., and Jaiman, R. K., 2017. "An Efficient Deep Learning Technique for the Navier-Stokes Equations: Application to Unsteady Wake Flow Dynamics".

[24] Hajgató, G., Gyires-Tóth, B., and Paál, G., 2019. "Accelerating Convergence of Fluid Dynamics Simulations with Convolutional Neural Networks". *Periodica Polytechnica Mechanical Engineering,* **63**(3), pp. 230–239.

[25] Stoecklein, D., Lore, K. G., Davies, M., Sarkar, S., and Ganapathysubramanian, B., 2017. "Deep Learning for Flow Sculpting: Insights into Efficient Learning using Scientific Simulation Data". *Scientific Reports,* **7**(March), pp. 1–11.

[26] Sekar, V., Zhang, M., Shu, C., and Khoo, B. C., 2019. "Inverse design of airfoil using a deep convolutional neural network". *AIAA Journal,* **57**(3), pp. 993–1003.

[27] Mukherjee, R., Li, Q., Chen, Z., Chu, S., and Wang, H., 2018. "NeuralDrop: DNN-based Simulation of Small-Scale Liquid Flows on Solids".

[28] Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B., 2019. "Deep Fluids: A Generative Network for Parameterized Fluid Simulations". *Computer Graphics Forum,* **38**(2), pp. 59–70.

[29] Georgiou, T., Schmitt, S., Olhofer, M., Liu, Y., Back, T., and Lew, M., 2018. "Learning Fluid Flows". *Proceedings of the International Joint Conference on Neural Networks,* **2018-July**.

[30] Lee, S., and You, D., 2019. "Data-driven prediction of unsteady flow over a circular cylinder using deep learning". *Journal of Fluid Mechanics,* **879**(Schmid 2010), pp. 217–254.

[31] Orchard, A., Vogel, J., and Coffin, E. K., 2020. "A practical approach to sizing thermal protection for spaceshiptwo". In AIAA Scitech 2020 Forum, p. 0258.

[32] Medic, G., Kalitzin, G., and You, D., 2006. "Integrated RANS/LES computations of turbulent flow through a turbofan jet engine". *NASA CTR Annual Research Briefs*(2005), pp. 275–285.

[33] Min, A. T. W., Sagarna, R., Gupta, A., Ong, Y.-S., and Goh, C. K., 2017. "Knowledge transfer through machine learning in aircraft design". *IEEE Computational Intelligence Magazine,* **12**(4), pp. 48–60.

[34] Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N., 2017. "Exploring generalization in deep learning". In Advances in neural information processing systems, pp. 5947–5956.

[35] Hoffer, E., Hubara, I., and Soudry, D., 2017. "Train longer, generalize better: closing the generalization gap in large batch training of neural networks". In Advances in Neural Information Processing Systems, pp. 1731–

1741.

[36] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O., 2016. "Understanding deep learning requires rethinking generalization". *arXiv preprint arXiv:1611.03530*.

[37] Wang, H., Ren, K., and Song, J., 2017. "A closer look at batch size in mini-batch training of deep auto-encoders". In 2017 3rd IEEE International Conference on Computer and Communications (ICCC), IEEE, pp. 2756–2761.

[38] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P., 2016. "On large-batch training for deep learning: Generalization gap and sharp minima". *arXiv preprint arXiv:1609.04836*.

[39] Wu, L., Zhu, Z., et al., 2017. "Towards understanding generalization of deep learning: Perspective of loss landscapes". *arXiv preprint arXiv:1706.10239*.

[40] Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y., 2017. "Sharp minima can generalize for deep nets". *arXiv preprint arXiv:1703.04933*.

[41] Le, L., Patterson, A., and White, M., 2018. "Supervised autoencoders: Improving generalization performance with unsupervised regularizers". In Advances in Neural Information Processing Systems, pp. 107–117.

[42] Maurer, A., Pontil, M., and Romera-Paredes, B., 2016. "The benefit of multitask representation learning". *The Journal of Machine Learning Research,* **17**(1), pp. 2853–2884.

[43] Caruana, R., 1997. "Multitask learning". *Machine learning,* **28**(1), pp. 41–75.

[44] Rigol, J. P., Jarvis, C. H., and Stuart, N., 2001. "Artificial neural networks as a tool for spatial interpolation". *International Journal of Geographical Information Science,* **15**(4), pp. 323–343.

[45] Coelho, L. P., and Richert, W., 2015. *Building machine learning systems with Python*. Packt Publishing Ltd.

[46] Zhang, Y. "A Better Autoencoder for Image: Convolutional Autoencoder". pp. 1–7.

[47] Ioffe, S., and Szegedy, C., 2015. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". *arXiv preprint arXiv:1502.03167*.

[48] Sethian, J. A., 1996. "A fast marching level set method for monotonically advancing fronts". *Proceedings of the National Academy of Sciences of the United States of America,* **93**(4), pp. 1591–1595.