

## THE EFFECT OF LOSS FUNCTIONS ON THE DEEP LEARNING MODELING FOR THE FLOW FIELD PREDICTIONS AROUND AIRFOILS

Ali Dogan<sup>\*</sup>, Cihat Duru<sup>†</sup>, Hande Alemdar<sup>‡</sup>  
and Ozgur Ugras Baran<sup>§</sup>  
Middle East Technical University  
Ankara, Turkey

### ABSTRACT

*CNNFOIL is a CNN-based machine learning tool that solves flow around the airfoil with a machine learning methodology. CNNFOIL, which is being developed by our research group, can predict flowfield around airfoils from different families at transonic regimes. We have improved the training process and accuracy of the CNNFOIL solver by implementing new loss functions. In this study, the effects of an  $L_2$  -based loss function, a physics-informed loss function based on continuity equation and a gradient difference loss function on the flow field predictions around airfoils are investigated. The loss functions are implemented into an encoder-decoder based convolutional neural network model. The neural network model is trained with Reynolds-averaged Navier-Stokes (RANS) based computational fluid dynamics (CFD) simulation results for different airfoil shapes at zero angle of attack for 0.7 Mach number flow. Numerical experiments are carried out with an unseen airfoil shape to assess the effects of loss functions. The performance of each loss-functions are discussed.*

### INTRODUCTION

Many deep neural network models are introduced for computationally expensive fluid flow solutions with the recent advancements in machine learning. In particular, deep learning models make it possible to obtain accurate enough results quickly. These estimations can be used during time-consuming optimization and design exploration studies. In recent years, several studies have focused on predicting the flow field around various objects under diverse flow conditions. Guo et al. [Guo et al., 2016] achieved significant speedup with a convolutional neural network (CNN) model for the prediction of velocity fields over 2D and 3D domains. Jin et al. [Jin et al., 2018] proposed a CNN model to predict velocity fields around a cylinder for low Reynolds number flow. An encoder-decoder CNN model was developed by Bhatnagar et al. [Bhatnagar et al., 2019] for the prediction of velocity and pressure fields around airfoils. Another CNN model based on U-net architecture was proposed by Chen et al. [Chen et al., 2019] for the flow field prediction around various arbitrary shapes in laminar

---

<sup>\*</sup>Computer Engineering Department, Email: ali.dogan\_04@metu.edu.tr

<sup>†</sup>Mechanical Engineering Department, Email: cduru@metu.edu.tr

<sup>‡</sup>Computer Engineering Department, Email: alemdar@metu.edu.tr

<sup>§</sup>Mechanical Engineering Department, Email: ubaran@metu.edu.tr

flow. Several models based on generative adversarial network (GAN) and CNN were proposed by Lee et al. [Lee and You, 2019] in order to study unsteady flow field prediction around a circular cylinder. Sekar et al. [Sekar et al., 2019] studied the prediction of laminar flow fields around airfoils by utilizing both the CNN and the multilayer perceptron models. Another U-net architecture-based CNN model was proposed by Thuerey et al. [Thuerey et al., 2020] to predict flow fields around airfoils for incompressible flows. Apart from flow fields, deep learning methods are also utilized to predict airfoil lift coefficients. [Zhang et al., 2017] proposed several CNN models to predict the lift coefficient of a given airfoil shape and the flow conditions. Obtained results demonstrate that both MLP and CNN models achieve significant results in the prediction task.

Some of the proposed models use complex loss functions [Lee and You, 2019], consisting of physical law constraints or penalizing the gradient differences [Lee and You, 2019; Bhatnagar et al., 2019], whereas some models achieve comparable results using only a single  $L_2$ -based loss function [Guo et al., 2016; Jin et al., 2018; Chen et al., 2019; Sekar et al., 2019] or a simple  $L_1$  loss [Thuerey et al., 2020]. In the perspective of a deep learning practitioner, the prediction of flow fields around an object by utilizing a data-driven model is a regression problem. The main goal of a machine learning algorithm is to minimize the output of the loss function that measures the difference between the model output and the ground truth. As a consequence of the loss function minimization, the learnable parameters of the model are set optimum values. In regression tasks, bilateral loss functions such as  $L_1$  and  $L_2$  loss functions are shown to be the most suitable ones, rather than unilateral loss functions such as Hinge loss or logistic loss, which are more suitable in classification tasks [Nie et al., 2018].

In this study, we aim to investigate the implications of different loss functions in a flow field prediction task by focusing on the following non-mutually exclusive three questions:

1. Do models with  $L_2$ -based loss functions achieve significant improvements by combining the loss function with another?
2. Does a physics-informed loss function, i.e., conservation of mass, provide meaningful information to the models? Or is this implicitly embedded in  $L_2$ -based loss function?
3. Does penalizing the gradient differences in the flow field by a loss function, i.e., GDL, provide other benefits than edge smoothing to the model? Is it necessary to have a specific loss function in the flow field prediction problem just for this purpose?

In this study, we discuss these three questions by conducting experiments on our previously proposed encoder-decoder-based convolutional neural network model, CNNFOIL [Duru et al., 2021] for the prediction of flow fields around airfoils.

## METHOD

Numerical experiments are carried out to investigate the effect of loss functions on the flow field prediction around airfoils. Loss functions are implemented into the previously proposed model, CNNFOIL [Duru et al., 2021]. CNNFOIL is an encoder-decoder CNN model that aims to predict pressure coefficient fields around airfoils for 0.7 Mach number flow at zero angle of attack. The encoder part of the model consists of 8 convolutional layers with 2x2 filter size and a stride of 2, along with batch normalization layer and ELU (exponential linear unit) activation unit except the last layer. The decoder part consists of 8 transposed convolution operations having the same filter size and stride as encoder part, along with the ELU activation unit except for the last layer. Further details regarding the CNNFOIL can be found in [Duru et al., 2021].

In the present study, CNNFOIL is trained with the solutions of Reynolds-averaged Navier-Stokes (RANS) based computational fluid dynamics (CFD) simulations for 204 different airfoil shapes in order to estimate density,  $x$  and  $y$  components of the velocity,  $(u, v)$  around airfoils. An in-house finite-volume CFD solver is utilized to solve the compressible RANS equations. The Spalart-Allmaras turbulence model is used as the turbulence model. Viscous fluxes are evaluated using a second-order

central discretization scheme. A very accurate second-order HLLC flux scheme [Toro et al., 1994] is employed for inviscid fluxes. The second-order accuracy is achieved by the application of the Venkatakrishnan limiter [Venkatakrishnan, 1995]. A backward Euler time integration method with local time-stepping is employed for the temporal discretization. The generated database is divided into three sets: 149, 29 and 26 for training, validation and test sets, respectively.

Three loss functions, namely  $L_2$ ,  $L_c$  and  $L_{gdl}$ , are used in the study. The contribution of each loss function to the base model is investigated by weighted combinations of  $L_c$  and  $L_{gdl}$  to the overall loss function. The base model is trained with only  $L_2$  loss function which can be expressed as follows:

$$L_2 = \sum_{n=1}^N [(\rho_n - \hat{\rho}_n)^2 + (u_n - \hat{u}_n)^2 + (v_n - \hat{v}_n)^2] \quad (1)$$

where  $\rho$ ,  $u$ ,  $v$  are the ground truth of density,  $x$ -component velocity and  $y$ -component of velocity, and  $\hat{\rho}$ ,  $\hat{u}$ ,  $\hat{v}$  is the model output of those flow variables.  $N$  is the number of data points in the flow field.

$L_c$  loss function applied in the present study is derived from the continuity equation.  $L_c$  can be defined as follows

$$L_c = \sum_i \sum_j |(\rho_e u_e - \rho_w u_w) - (\hat{\rho}_e \hat{u}_e - \hat{\rho}_w \hat{u}_w)| + |(\rho_n u_n - \rho_s u_s) - (\hat{\rho}_n \hat{u}_n - \hat{\rho}_s \hat{u}_s)| \quad (2)$$

Notice that, each pixel of the solution image is treated as a conservation volume. For a cell centered approach, we need to interpolate for the face values for density and velocity. Flow variables on each cell face are estimated by averaging the corresponding values from the two opposite side of the cell face as

$$\begin{aligned} \Psi_e &= (\Psi(i, j) + \Psi(i + 1, j))/2 & \Psi_w &= (\Psi(i - 1, j) + \Psi(i, j))/2 \\ \Psi_n &= (\Psi(i, j) + \Psi(i, j + 1))/2 & \Psi_s &= (\Psi(i, j - 1) + \Psi(i, j))/2 \end{aligned} \quad (3)$$

where  $(i, j)$  denotes the cell indexes on the Cartesian grid. The subscripts (e,w,n,s) indicate the cardinal points of a cell on a 2D plane.

Gradient Difference Loss,  $L_{gdl}$  is proposed by [Mathieu et al., 2015], penalizes the gradient differences and provides a smoother image in the predictions is defined by

$$L_{gdl}(Y, \hat{Y}) = \sum_{i,j} ||Y_{i,j} - Y_{i-1,j}| - |\hat{Y}_{i,j} - \hat{Y}_{i-1,j}||^\alpha + ||Y_{i,j-1} - Y_{i,j}| - |\hat{Y}_{i,j-1} - \hat{Y}_{i,j}||^\alpha \quad (4)$$

where  $Y = \{\rho, u, v\}$  and  $\hat{Y} = \{\hat{\rho}, \hat{u}, \hat{v}\}$  are the ground truth and prediction of flow variables, respectively. Parameter  $\alpha$  is chosen to be 1 in all experiments.

In our experiments, the loss function is defined as the weighted combination of loss functions mentioned before. The obtained loss function is given as

$$Loss(Y, \hat{Y}) = \lambda_2 L_2 + \lambda_c L_c + \lambda_{gdl} L_{gdl} \quad (5)$$

The experimental matrix is listed in Table 1. In order to adapt the order of magnitudes of the loss functions,  $\lambda_c$  is chosen 100 in corresponding experiments. All three flow field channels are normalized before passing through the neural network. However, de-normalization is applied before the loss function calculation step to prevent incorrect back propagation due to the multiplication of different channels in the loss function (e.g.,  $\rho_e u_e$ ).

| Experiment            | $\lambda_2$ | $\lambda_c$ | $\lambda_{gdl}$ |
|-----------------------|-------------|-------------|-----------------|
| $L_2$                 | 1           | 0           | 0               |
| $L_2 + L_c$           | 1           | 100         | 0               |
| $L_2 + L_{gdl}$       | 1           | 0           | 1               |
| $L_2 + L_c + L_{gdl}$ | 1           | 100         | 1               |

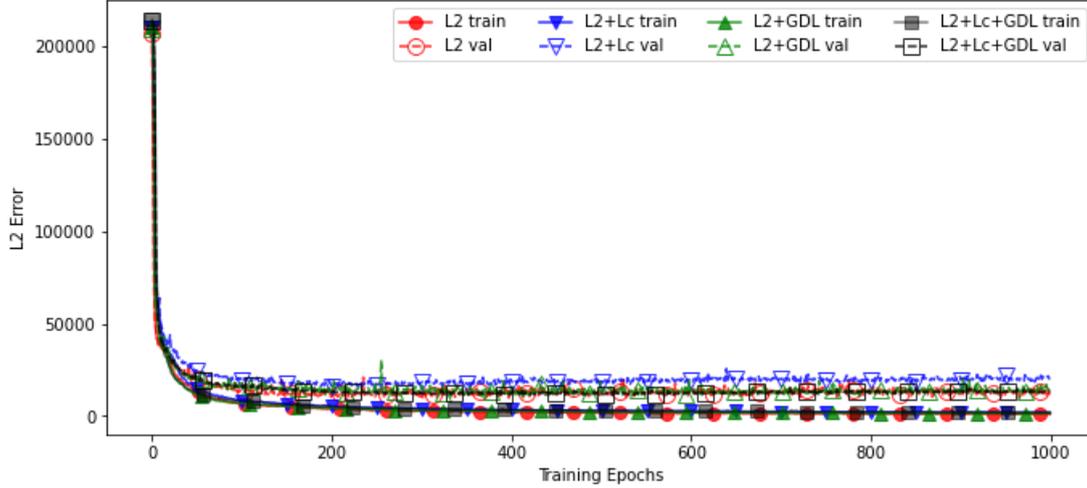


Figure 1: Convergence history of  $L_2$  error function

## RESULTS AND DISCUSSION

Training history of each case in Table 1 with respect to each error function given in Eq. 1, Eq. 2 and Eq. 4 are depicted in Fig. 1, Fig. 2 and Fig. 3, respectively. Each model is represented with a different color in the figures, as shown at the right top corner. Different shapes are embedded into the lines of each model to prevent confusion where the lines overlap.

Error functions inspection for each experiment is vital because error functions show the behavior of the model under the applied loss functions. For example, all models should exhibit convergence in  $L_2$  error since it is the base loss function in all experiments. Moreover, we expect to see relatively good convergence in  $L_c$  error in all models whether or not the model has  $L_c$  loss function since it indicates the physical conservation law of mass. In fact, we see this in Fig. 2 where  $L_c$  error decreases regardless of the existence of  $L_c$  loss function.

Fig. 1 shows that combining  $L_2$  loss function with  $L_c$  or  $L_{gdl}$  loss functions does not provide a strong contribution to minimizing the  $L_2$  error function. However, implementing  $L_c$  or  $L_{gdl}$  loss functions into  $L_2$  loss function have significant effect on the convergence of mass conservation as can be seen in Fig. 2. Even, just adding the  $L_{gdl}$  loss function may help to satisfy mass conservation than  $L_2$  loss function by itself. Fig. 3 shows the convergence history of  $L_{gdl}$  during the training of each case in Table 1. Fig. 3 demonstrates better convergence history of  $L_{gdl}$  for the cases in which the  $L_{gdl}$  loss function appeared. However, adding  $L_c$  loss function has an adverse effect on the  $L_{gdl}$  errors by resulting in much more  $L_{gdl}$  error than  $L_2$  loss function by itself.

Furthermore, density,  $u$  and  $v$  velocity fields of an airfoil from the test set are compared for each case in Table 1 in Fig. 4, 5 and 6, respectively. The ground truth, the prediction and the absolute error between the ground truth and the prediction for the density,  $u$  and  $v$  velocity fields are plotted in Fig. 4, 5 and 6. Even though combining  $L_c$  and/or  $L_{gdl}$  loss functions with  $L_2$  loss function improves  $L_c$  and  $L_{gdl}$  errors as shown in Fig. 2 and 3, the loss combination cases do not improve predictions significantly compared to the case in which only  $L_2$  loss function is included. This may

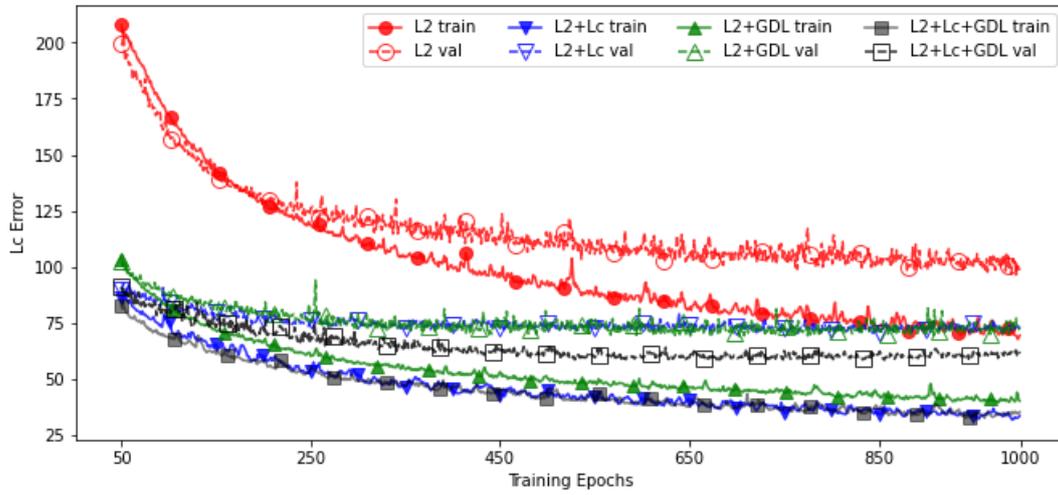


Figure 2: Convergence history of  $L_c$  error function

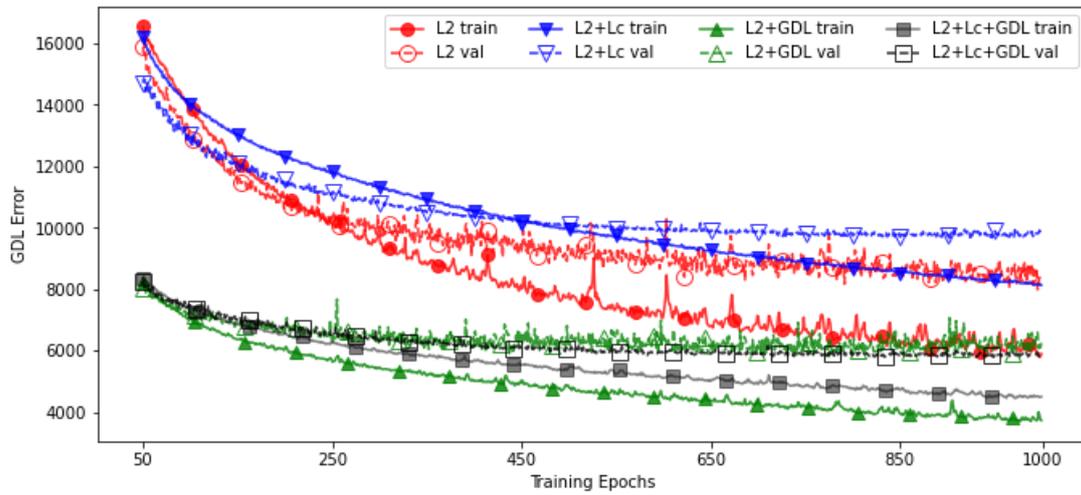


Figure 3: Convergence history of  $L_{gdl}$  error function

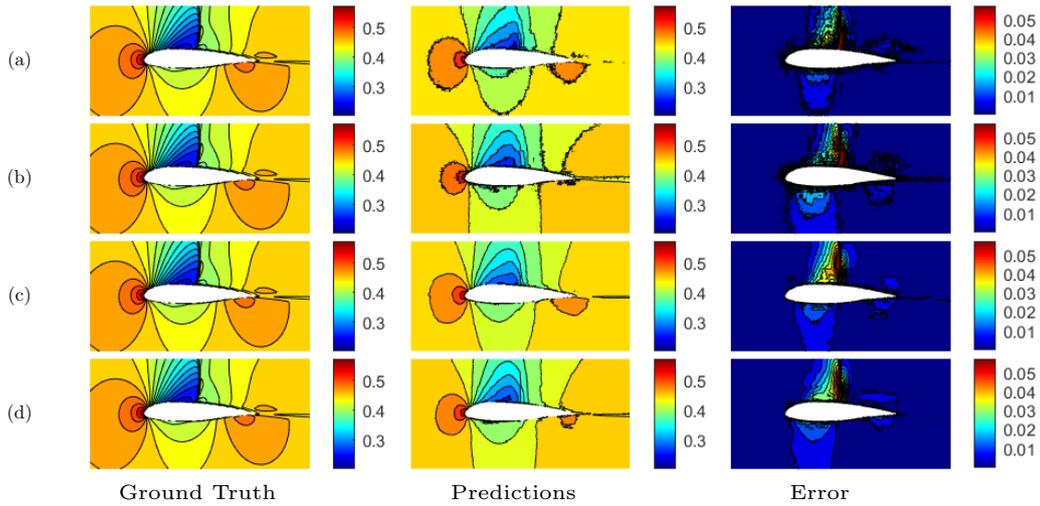


Figure 4: Density fields of LS 417 airfoil from the test set. Left column: ground truth extracted from CFD simulations, Middle column: the model output, Right Column: the absolute difference of density fields between the model output and the ground truth for (a)  $L_2$ , (b)  $L_2 + L_c$ , (c)  $L_2 + L_{gdl}$ , (d)  $L_2 + L_c + L_{gdl}$ .

also demonstrate that  $L_2$  error function is enough to meet the convergence criteria for the training of a deep learning model. However, we can observe smooth contour lines in the cases in which  $L_{gdl}$  loss function appeared and the adverse effect of  $L_c$  loss function on the  $L_{gdl}$  errors mentioned before. Apart from the loss function comparison, the model performs reasonably well in the flow field prediction for three flow variables except for the shock waves, as can be seen in the absolute errors in Fig. 4, 5 and 6.

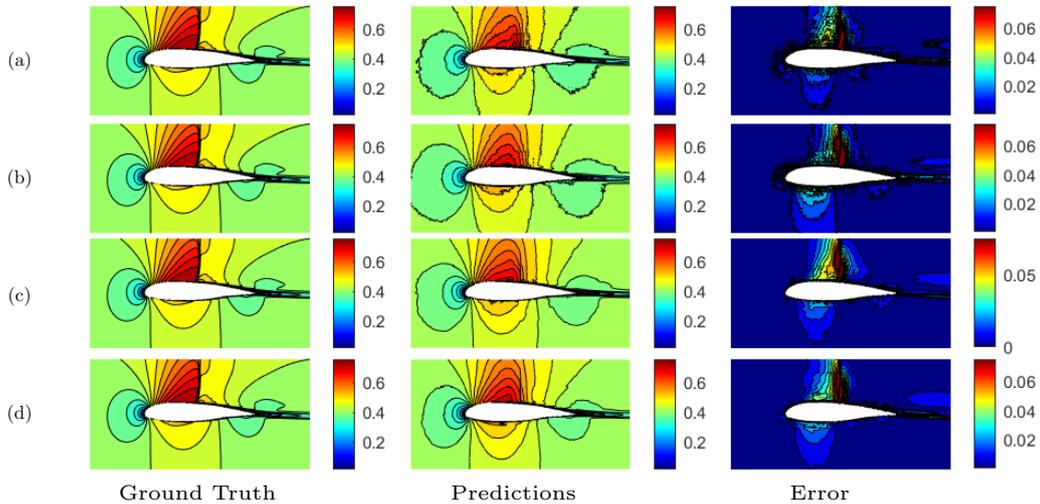


Figure 5:  $x$ -component velocity fields of LS 417 airfoil from the test set. Left column: ground truth extracted from CFD simulations, Middle column: the model output, Right Column: the absolute difference of density fields between the model output and the ground truth for (a)  $L_2$ , (b)  $L_2 + L_c$ , (c)  $L_2 + L_{gdl}$ , (d)  $L_2 + L_c + L_{gdl}$ .

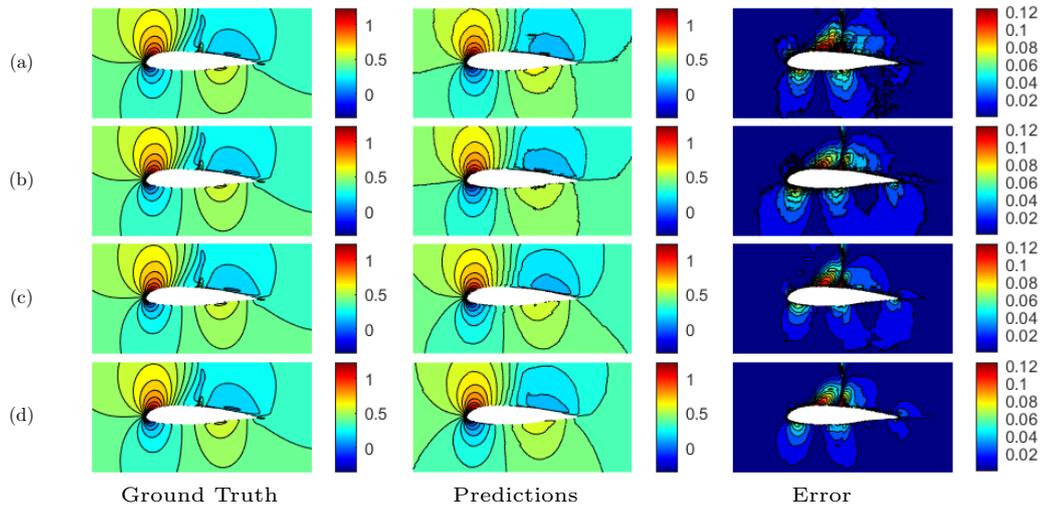


Figure 6:  $y$ -component velocity fields of LS 417 airfoil from the test set. Left column: ground truth extracted from CFD simulations, Middle column: the model output, Right Column: the absolute difference of density fields between the model output and the ground truth for (a)  $L_2$ , (b)  $L_2 + L_c$ , (c)  $L_2 + L_{gdl}$ , (d)  $L_2 + L_c + L_{gdl}$ .

## CONCLUSIONS

In this study, we investigated the role of loss functions utilized in deep neural network models for fluid flow studies. The experiments with different loss functions are conducted by implementing them on the previously proposed CNNFOIL model for estimating flow fields around airfoils. For the first question we had asked at the beginning, initial findings show that combining physics-informed loss function based on the conservation of mass and gradient difference loss with an  $L_2$  based loss function did not improve the flow field predictions significantly in terms of  $L_2$  error. However, the loss combinations helped to satisfy the conservation of mass and edge smoothing in the flow fields. This improvement shows that physic-informed loss functions do provide additional information along with  $L_2$ -based loss function, which answers our second question. For the last question, we see that GDL provides other benefits than edge smoothing to the model. We see that it helps to the mass of conservation almost as much as  $L_c$  loss does. A comprehensive investigation will be carried out by also examining the conservation of momentum and energy. This could help physical loss function to provide more information to the model and may lead to better results. Furthermore, conducting experiments on other neural network models to compare the loss functions can help to see more specific results and generalize the study better.

## References

- Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., and Kaushik, S. (2019) *Prediction of aerodynamic flow fields using convolutional neural networks*, Springer, Vol 64, p: 525-545, May 2019
- Chen, J., Viquerat, J., and Hachem, E. (2019) *U-net architectures for fast prediction of incompressible laminar flows*, arXiv preprint arXiv:1910.13532, Oct 2019
- Duru, C., Alemdar, H., and Baran, O. U. (2021) *CNNFOIL: convolutional encoder decoder modeling for pressure fields around airfoils*, Springer, Vol 33, p: 6835-6849, Jun 2021
- Guo, X., Li, W., and Iorio, F. (2016) *Convolutional neural networks for steady flow approximation*, ACM, p: 482-490, Aug 2016
- Jin, X., Cheng, P., Chen, W.-L., and Li, H. (2018) *Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder*, AIP Publishing LLC, Vol 30, p: 047105, Apr 2018
- Lee, S. and You, D. (2019) *Data-driven prediction of unsteady flow over a circular cylinder using deep learning*, Cambridge University Press, Vol 879, p: 217-254, Sep 2019
- Mathieu, M., Couprie, C., and LeCun, Y. (2015) *Deep multi-scale video prediction beyond mean square error*, arXiv preprint arXiv:1511.05440, Nov 2015
- Nie, F., Zhanxuan, H., and Li, X. (2018) *An investigation for loss functions widely used in machine learning*, Communications in Information and Systems, Vol 18. p:37-52, Jan 2018
- Sekar, V., Jiang, Q., Shu, C., and Khoo, B. C. (2019) *Fast flow field prediction over airfoils using deep learning approach*, AIP Publishing, Vol 31, p: 057103, May 2019
- Thuerey, N., Weißenow, K., Prantl, L., and Hu, X. (2020) *Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows*, American Institute of Aeronautics and Astronautics, Vol 58, p:25-36, Oct 2020
- Toro, E. F., Spruce, M., and Speares, W. (1994) *Restoration of the contact surface in the HLL-Riemann solver*, Shock Waves, Vol 4, p: 25-34, Jul 1994
- Venkatakrishnan, V. (1995) *Convergence to steady state solutions of the Euler equations on unstructured grids with limiters*, Elsevier, Vol 118, p: 120-130, Apr 1995
- Zhang, Y., Sung, W., and Mavris, D. (2017) *Application of Convolutional Neural Network to Predict Airfoil Lift Coefficient*, ArXiv, Vol abs/1712.10082, Jan 2018