Naval Surface Warfare Center
Dahlgren Division

Toward the use of convolutional neural networks as a post-processing enhancement to RANS-modeled turbulence

Presented by

Alec Brodeur
Engineer, Hypersonic Design Integration & Analysis Branch
Naval Surface Warfare Center Dahlgren Division

John Romano
Head, Hypersonic Design, Integration & Analysis Branch
Naval Surface Warfare Center Dahlgren Division

July 28, 2022

The Leader in Warfare Systems Development and Integration

Agenda

- Introduction
- Theory
  - Datasets
  - Computational Fluid Dynamics (CFD) Modeling
  - Convolutional Neural Networks (CNN) Modeling
  - Hyperparameter Optimization
- Results and Discussion
- Continuing Work

Introduction

- Increased computational power has enabled meaningful advances in machine learning (ML) and the prevalence of high-fidelity, CFD simulation datasets
- Modest progress made toward leveraging ML models to improve predictions of flow-field behavior computationally[1] and to more accurately measure fluid phenomena experimentally [2]–[4]
  1. To increase simulation fidelity by using ML to augment solution algorithms and physical models (such as those describing turbulent behavior in Reynolds-Averaged Navier-Stokes (RANS) simulations), and
  2. To reduce solution runtimes by leveraging reduced-order modeling or super-resolution techniques

1. K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence Modeling in the Age of Data," pp. 1–23, 2019.
2. S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine Learning for Fluid Mechanics," *Annu. Rev. Fluid Mech.*, vol. 52, no. 1, pp. 477–508, 2020.
3. K. Taira *et al.*, "Modal analysis of fluid flows: Applications and outlook," *AIAA J.*, vol. 58, no. 3, pp. 998–1022, 2020.
4. N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz, "Shallow neural networks for fluid flow reconstruction with limited sensors," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 476, no. 2238, p. 20200097, 2020.

Objective

- Previous study[5] extended existing literature[6] in reduced-order modeling and considered CNN models predicting *high-accuracy* vorticity fields from *low-accuracy* vorticity fields for transonic, 4-digit NACA airfoils at high angles of attack
  - Results showed promise but required further improvements and generalizations
- Recent efforts studied other field variables and increased predictive capability
- Improvements realized by hyperparameter optimization

5. J. Romano and O. Baysal, “Convolutional-neural-network-based Auto-encoder for Synthetic Upscaling of Computational Fluid Dynamics Simulations.” *AIAA SCITECH 2022 Forum*, p. 0186. 2022.

6. L. Agostini, “Exploration and prediction of fluid dynamical systems using auto-encoder technology,” *Phys. Fluids*, vol. 32, no. 6, p. 067103, 2020.

Theory – CFD Modeling

- NASA's Fully Unstructured Navier-Stokes 3D (FUN3D) generated the simulations
  - Transient calculations with 1st order temporal discretizations, 75 subiterations, and max CFL of 10
  - Closure from SA model for URANS and SA-based DES formulations
  - 500 start-up time-steps resolved start-up transients, followed by a 100 time-step sampling period
  - Separate computational grids for URANS and DES calculations
- CFD simulation data linearly interpolated onto a Cartesian grid for CNN processing
  - Datasets replicated with different scales and translations to create larger datasets for training and testing
  - 176 x 512 (H x W) data points

Table 1: CFD grid metrics

for chord length *c*	URANS Grid	DES Grid
Farfield radius	10*c*	10*c*
Span	0.06*c*	0.06*c*
Airfoil circumferential partitions	550	2,500
Radial partitions	125	500
Spanwise partitions	10	60
Wall initial cell height	5e-5*c*	1e-5*c*

Theory: Data Preprocessing

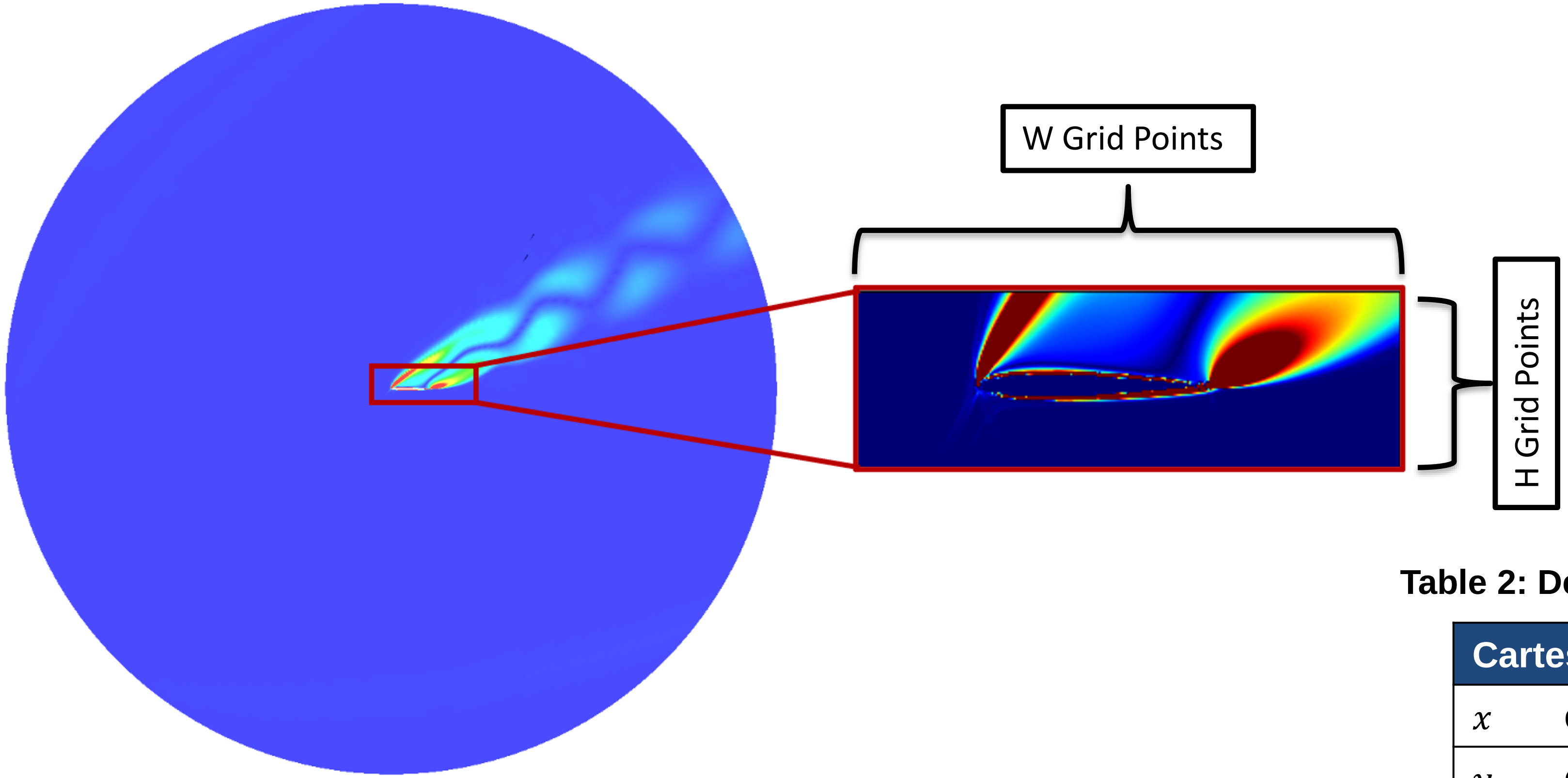


Fig. 1: Data preprocessing grid transformation schematic

Table 2: Description of dataset preprocessing

Cartesian grid coordinate extents	
x	$\in [(-0.5+t)sc, (2.5+t)sc]$
y	$\in [-0.3sc, 0.3sc]$
s	$\in [0.8, 0.9, 1, 1.1, 1.2]$
t	$\in [-0.2, -0.1, 0, 0.1, 0.2]$
Cartesian grid dimension	
H	176
W	512

Theory – Datasets

- Time-averaged unsteady RANS (*low-accuracy*) and DES (*high-accuracy*) CFD calculations used as training and testing datasets
- All cases run at Mach 0.728 with sea-level atmospheric conditions
- Vorticity magnitude, density, and pressure ere flow-field variables for the current study
- Two studies considered sensitivity to
  - Variation in geometry (NACA0006, NACA0012, NACA2412, NACA4412), and
  - Variation in angle of attack (NACA0006 at $\alpha = -30 \text{ to } 30$)
- This presentation focuses on results for the angle of attack study with pressure

Table 3: Description of datasets

Study	Training Dataset	Testing Dataset
α Sensitivity	NACA0006 $\alpha \in [-30°, -10°, 0°, 10°, 30°]$	NACA0006 $\alpha \in [-20°, 20°, 25°]$
Geometry Sensitivity	NACA0006 $\alpha \in [20°, 30°]$ NACA0012 $\alpha \in [20°, 30°]$ NACA4412 $\alpha \in [20°, 30°]$	NACA0006 $\alpha = 25°$ NACA0012 $\alpha = 25°$ NACA4412 $\alpha = 25°$ NACA2412 $\alpha \in [20°, 25°, 30°]$

Theory – CNN Modeling and Optimization

- Separate CNN autoencoder networks generated for each study
- Questions about network shape in SciTech led to the idea that the shape (and other hyperparameters) should be solved by an optimizer
- Used the Sequential Model-based Algorithm Configuration (SMAC) Python library[7] to optimize hyperparameters
  - Random forest search determined optimal configuration from predefined search space
- Considered the broad search space in Table 4

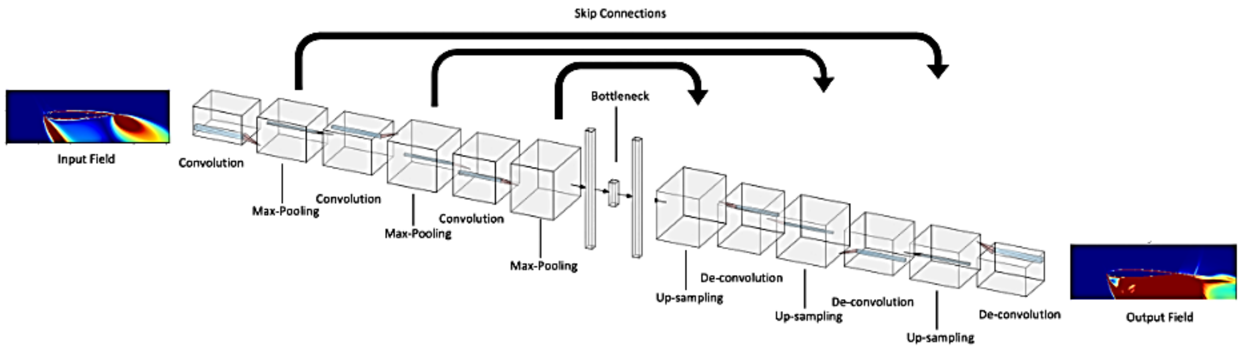


Fig. 2: Representative CNN autoencoder model schematic

Table 4: Hyperparameter search space

Hyperparameter	Type	Search Space	Default Value (SciTech 2022)	Optimal (based on loss function)
Convolutional Filters	Integer	[10:100]	48	35
Activation Function	Category	sigmoid relu elu tanh selu	sigmoid	tanh
Filter/Pooling Kernel Size	Category	[2,4,8]	2	2
Number of Convolution Layers	Integer	[1:4]	3	2
Network Optimizer	Category	adam adadelta adagrad adamax nadam	Adadelta	adam
Loss Function	Category	mse mae mape msle	msle	msle
Latent Space Dimension	*Integer*	*[5:50]*	12	46

7. F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond Manual Tuning of Hyperparameters," *KI - Kunstl. Intelligenz*, vol. 29, no. 4, pp. 329–337, 2015.

Model and Training History Comparison

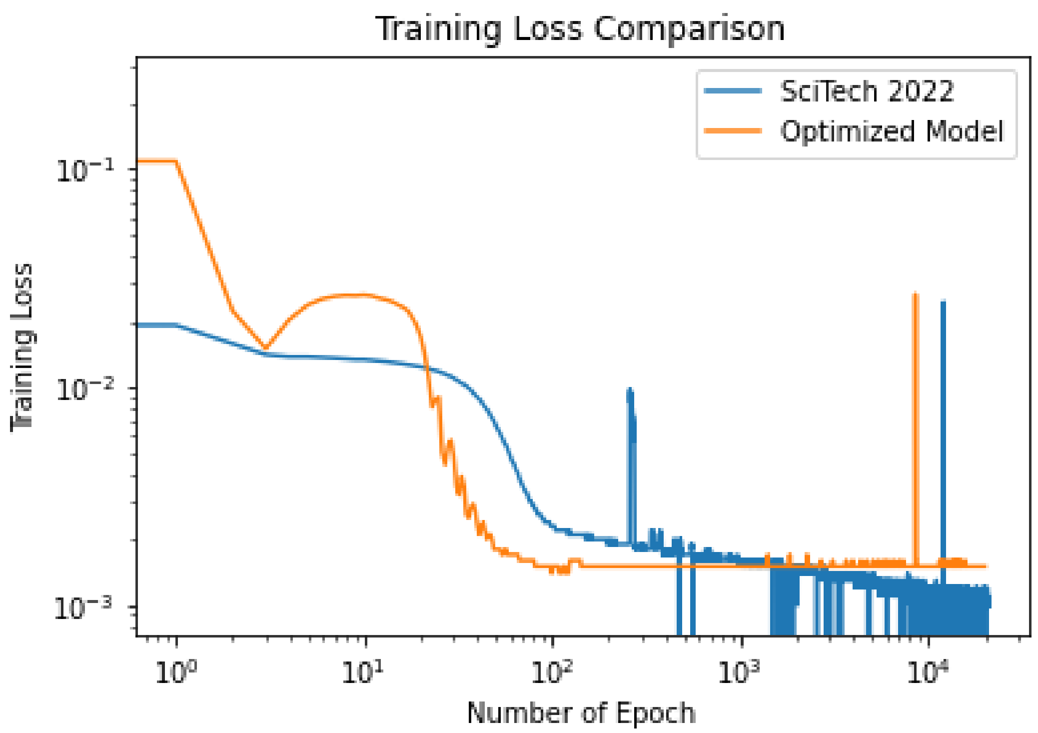


Fig. 3: Training histories for SciTech 2022 and optimized models

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input (InputLayer)	[(None, 176, 512, 1,	0	
batch_normalization (BatchNorma	(None, 176, 512, 1,	4	input[0][0]
noise (GaussianNoise)	(None, 176, 512, 1,	0	batch_normalization[0][0]
conv3d (Conv3D)	(None, 176, 512, 1,	1248	noise[0][0]
max_pooling3d (MaxPooling3D)	(None, 88, 256, 1, 4	0	conv3d[0][0]
conv3d_1 (Conv3D)	(None, 88, 256, 1, 9	41568	max_pooling3d[0][0]
max_pooling3d_1 (MaxPooling3D)	(None, 44, 128, 1, 9	0	conv3d_1[0][0]
conv3d_2 (Conv3D)	(None, 44, 128, 1, 1	166080	max_pooling3d_1[0][0]
max_pooling3d_2 (MaxPooling3D)	(None, 22, 64, 1, 19	0	conv3d_2[0][0]
flatten (Flatten)	(None, 270336)	0	max_pooling3d_2[0][0]
bottleneck (Dense)	(None, 12)	3244044	flatten[0][0]
dense (Dense)	(None, 270336)	3514368	bottleneck[0][0]
reshape (Reshape)	(None, 22, 64, 1, 19	0	dense[0][0]
concatenate (Concatenate)	(None, 22, 64, 1, 38	0	reshape[0][0] max_pooling3d_2[0][0]
up_sampling3d (UpSampling3D)	(None, 44, 128, 1, 3	0	concatenate[0][0]
conv3d_transpose (Conv3DTranspo	(None, 44, 128, 1, 1	663744	up_sampling3d[0][0]
concatenate_1 (Concatenate)	(None, 44, 128, 1, 2	0	conv3d_transpose[0][0] max_pooling3d_1[0][0]
up_sampling3d_1 (UpSampling3D)	(None, 88, 256, 1, 2	0	concatenate_1[0][0]
conv3d_transpose_1 (Conv3DTrans	(None, 88, 256, 1, 9	248928	up_sampling3d_1[0][0]
concatenate_2 (Concatenate)	(None, 88, 256, 1, 1	0	conv3d_transpose_1[0][0] max_pooling3d[0][0]
up_sampling3d_2 (UpSampling3D)	(None, 176, 512, 1,	0	concatenate_2[0][0]
conv3d_transpose_2 (Conv3DTrans	(None, 176, 512, 1,	172848	up_sampling3d_2[0][0]
output (Conv3DTranspose)	(None, 176, 512, 1,	49	conv3d_transpose_2[0][0]

Total params: 8,052,881
Trainable params: 8,052,879
Non-trainable params: 2

Fig. 4: SciTech 2022 Model Description

Model: "model"

Layer (type)	Output Shape	Param #
in (InputLayer)	[(None, 176, 512, 1, 1)]	0
batch_normalization (BatchNo	(None, 176, 512, 1, 1)	4
conv3d (Conv3D)	(None, 176, 512, 1, 35)	175
max_pooling3d (MaxPooling3D)	(None, 88, 256, 1, 35)	0
batch_normalization_1 (Batch	(None, 88, 256, 1, 35)	140
conv3d_1 (Conv3D)	(None, 88, 256, 1, 70)	9870
max_pooling3d_1 (MaxPooling3	(None, 44, 128, 1, 70)	0
flatten (Flatten)	(None, 394240)	0
bottleneck (Dense)	(None, 46)	18135086
dense (Dense)	(None, 394240)	18529280
reshape (Reshape)	(None, 44, 128, 1, 70)	0
up_sampling3d (UpSampling3D)	(None, 88, 256, 1, 70)	0
conv3d_transpose (Conv3DTran	(None, 88, 256, 1, 70)	19670
up_sampling3d_1 (UpSampling3	(None, 176, 512, 1, 70)	0
conv3d_transpose_1 (Conv3DTr	(None, 176, 512, 1, 35)	9835
output (Conv3DTranspose)	(None, 176, 512, 1, 1)	36

Total params: 36,704,096
Trainable params: 36,704,024
Non-trainable params: 72

Fig. 5: Optimized Model Description

Compare SciTech 2022 and Optimized Models

- Model comparisons given for time-averaged pressure field for NACA0006 airfoil
- Optimized model does a better job predicting the DES pressure field for the test dataset but creates low-resolution predictions
- Revisiting optimizer run to improve prediction resolution
  - Possibly set accuracy rather than loss function as an optimization objective
  - Potentially add more layers into system

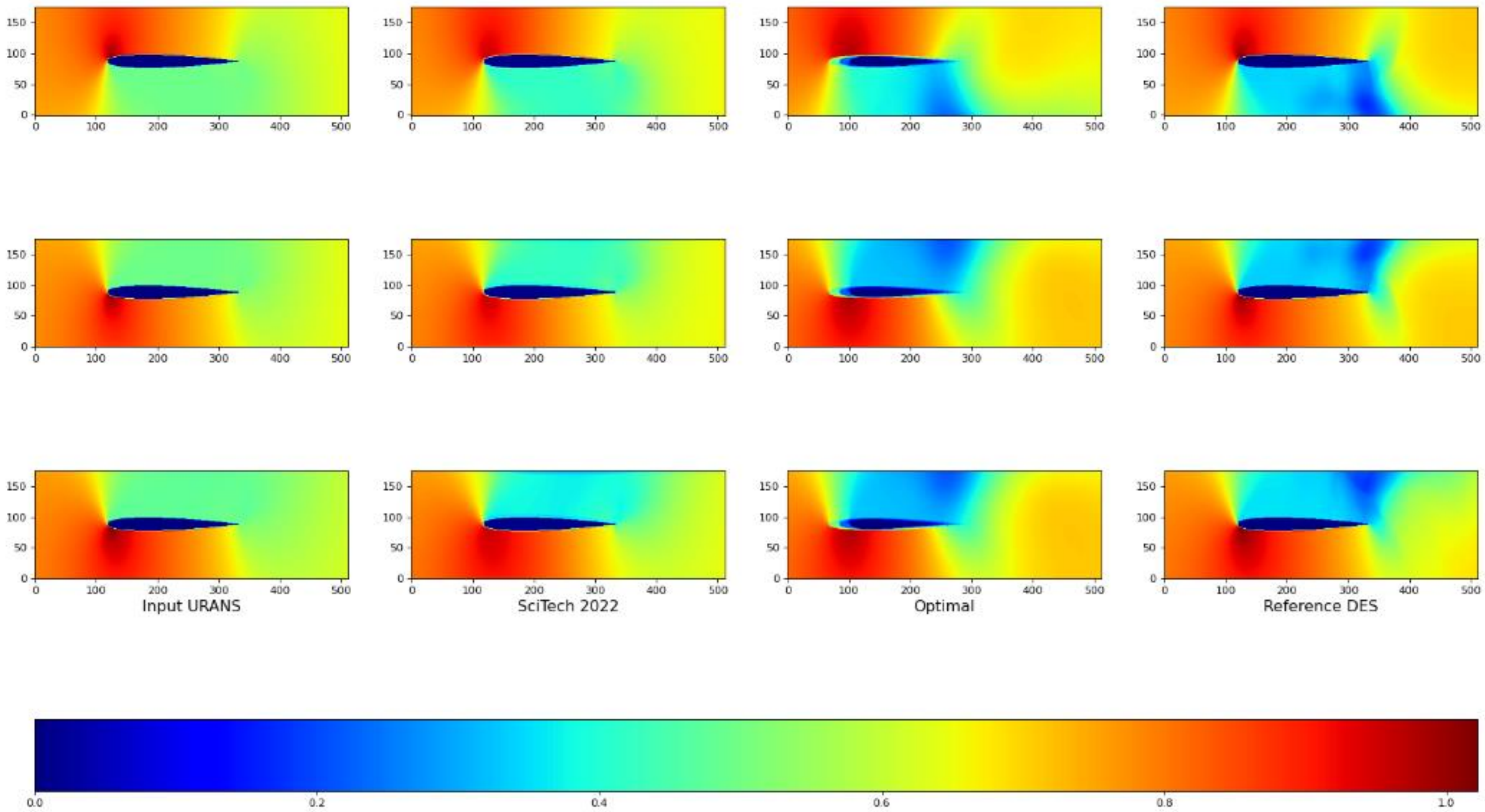
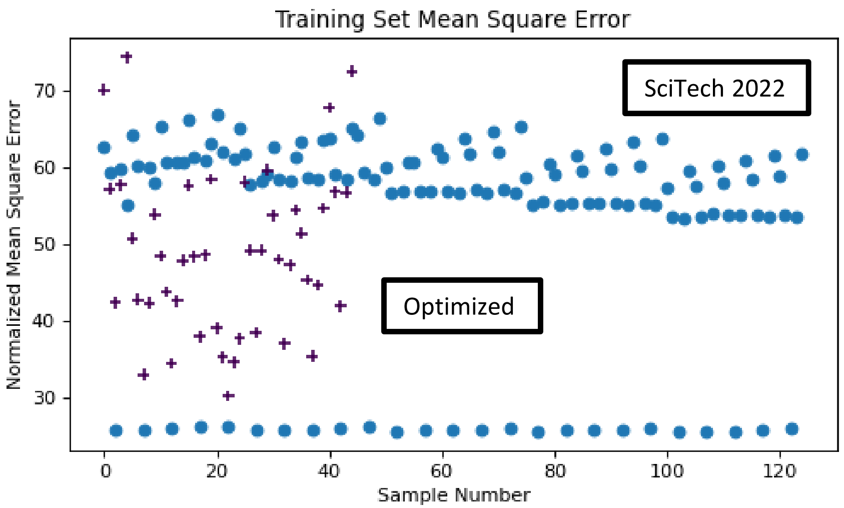


Figure 6: Qualitative nondimensional pressure field predictions

Results/Discussion – Quantitative Errors

- CNN model errors evaluated quantitatively based on mean square error
- Optimized model generally outperforms SciTech 2022 model across all angles of attack in testing and training sets



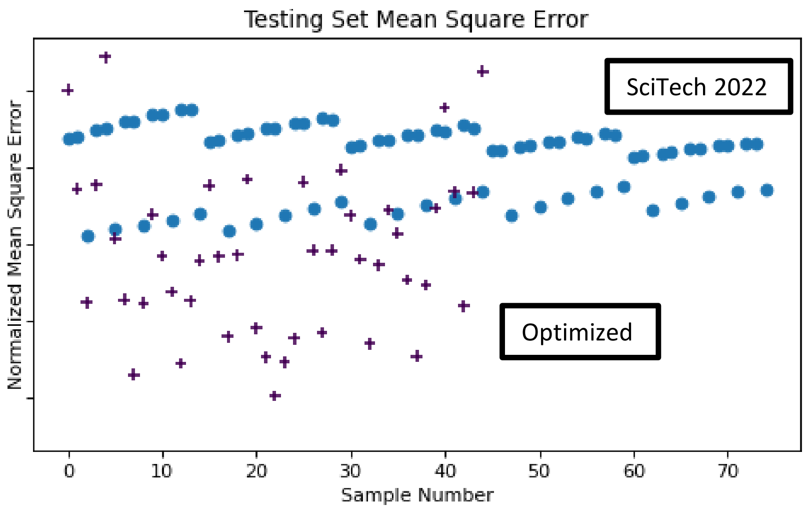


Figure 7: Mean square error comparison between training and testing datasets

Conclusions/Path Forward

- CNN autoencoder model constructed to generate *higher-accuracy* flow field predictions based on *lower-accuracy* flow field inputs
- Pressure field predictions improved after hyperparameter optimization
  - More work required to increase predicted field accuracy

- Next steps to continuously improve predictive capability
  - Continue work with hyperparameter optimization
  - Consider other network architectures, such as GAN and HRNet
  - Consider alternative data preprocessing approaches and increased training

Glossary

AE	autoencoder
CFD	computational fluid dynamics
CFL	Courant–Friedrichs–Lewy
CNN	convolutional neural networks
CPU	central processing unit
DES	detached-eddy-simulation
FUN3D	Fully Unstructured Navier-Stokes 3D
GAN	Generative Adversarial Network
GPU	graphics processing unit
HRNet	High-Resolution Network
hrs	hours
ML	machine learning
MSE	mean square error
NACA	National Advisory Committee for Aeronautics
NASA	National Aeronautics and Space Administration
RANS	Reynolds Average Navier-Stokes
SA	Spalart-Allmaras Model
SMAC	Sequential Model-based Algorithm Configuration
URANS	Unsteady Reynolds-Averaged Navier-Stokes

Acknowledgment

- Dr. Oktay Baysal from Old Dominion University provided guidance to this work as a member of Mr. Romano's Ph.D. advisory committee.

Thank You