Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.

Incremental Super-Resolution Reconstruction for Turbulent Flow on High-Order Discontinuous Finite Elements

Miles J. McGruder* and Krzysztof J. Fidkowski†
Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

Adjoint methods provide output based error indicators that enable large-scale adaptive simulations. However, the adjoint solution becomes unstable for large-scale chaotic flows, including scale-resolving simulations of turbulence. Recently, artificial reconstruction of turbulent flows has been used as the basis for an adaptive error indicator. In this work, we propose a residual artificial neural network that incrementally reconstructs a turbulent flow represented on discontinuous finite elements. We discuss the challenges of the finite element setting and show accurate spectral properties for reconstruction over a large order jump.

I. Introduction

Super-resolution has been an active area of computer vision research for several decades, typically operating on digital image data. The goal is to reconstruct a high-resolution output from a low-resolution or distorted input. The down-scaling is typically posed as

$$\mathbf{u}_H = \mathcal{M}\left(\mathbf{u}_h\right), \tag{1}$$

where $\mathcal{M}$ is a, generally unknown, down-scaling operator that reduces the high-resolution input $\mathbf{u}_h$ to the low resolution output $\mathbf{u}_H$. Here, $\mathbf{u}_h$ and $\mathbf{u}_H$ represent field quantities over the spatial domain, *e.g.* a velocity in a CFD simulation. In super-resolution we seek an up-scaling model $\mathcal{M}^{-1}$ that approximates the original image as

$$\hat{\mathbf{u}}_h = \mathcal{M}^{-1}\left(\mathbf{u}_H\right), \tag{2}$$

where $\hat{\mathbf{u}}_h$ is the reconstructed image. Super-resolution is an ill-posed inverse mapping problem. Since exact solutions are generally impossible, we seek a solution that minimizes up-scaling error. Classical algorithms dominated in super-resolution for decades [1, 2]. With recent advances in compute capability and data availability, artificial neural networks have surpassed the performance of classical algorithms for super-resolution [3, 4]. Hence, they will also be the focus of this paper.

Super-resolution is well-studied in the literature, with various network architectures tuned for different applications [3, 4]. A few of these architectures are most commonly used in super-resolution for fluids and are discussed here. Convolutional neural networks [5, 6] achieved state of the art up-scaling performance when Dong *et al.*[7] formulated classical ideas in the form of a single convolutional neural network. Ledig *et al.*[8] increased performance by applying a generative adversarial framework to the problem [9]. Previous super-resolution work [10, 11] has also used residual networks [12], where a network stage does not learn the final output, but a correction to the input. These residual networks serve as the primary inspiration for the work in this paper.

Super-resolution in a fluid dynamics context is typically applied not to a set of images, but to downsampled turbulent velocity fields. Fukami *et al.*[13] tested a hybrid convolutional neural network architecture on DNS data using max and average pooling for downsampling. Liu *et al.*[14] proposed an approach for spatio-temporal super-resolution using multiple convolutional paths, each handling different time ranges. Fukami *et al.*[15] proposed an alternative temporal super-resolution architecture built on their previous single-image work. Generative adversarial approaches have also been applied to super-resolution. Xie *et al.*[16] include an additional temporal discriminator during training to ensure the output's temporal coherence. Deng *et al.*[17] successfully resolve large wake fields with 4x and 8x upscaling factors using their generative adversarial approach.

Pradhan and Duraisamy [18] explored super-resolution for turbulent flows projected to discontinuous finite elements. They showed remarkable reconstruction accuracy at relatively low polynomial orders. Xu, Pradhan, and Duraisamy [19] augmented the previous model by altering a hidden layer's weights with a trainable function of an input parameter. The excellent reconstruction of the baseline model was further improved.

*Graduate Research Assistant
†Professor, AIAA Associate Fellow

For the remainder of the paper we will begin by motivating the work in Section II. Section III presents the network architectures tested and the generation of their training data. Section IV compiles and discusses super-resolution results across various order jumps. Finally, Section V discusses the bigger picture for this work and future directions.

II. Motivation

This work is part of a larger project to enable adaptation of chaotic turbulent flows using the discontinuous Galerkin (DG) finite element method. The governing equations for all training and testing data are the Navier-Stokes equations. Written in compact form they are

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}) - \nabla \cdot \vec{\mathbf{G}}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{0}, \tag{3}$$

where $\mathbf{u} \in \mathbb{R}^s$ is the rank s state vector, $\vec{\mathbf{F}}$ is the inviscid flux, and $\vec{\mathbf{G}}$ is the viscous flux.

DG splits a computational domain Ω into a tessellation $\mathcal{T}_h$ of non-overlapping elements, each covering a volume Ω_e. The state is represented as a linear combination of basis functions

$$\mathbf{u} = \sum_{e}^{N_e} \sum_{i}^{N_b} \mathbf{U}_{e,i} \phi_{e,i}, \tag{4}$$

where N_e is the number of elements, N_b is the number of basis functions on element e, $\mathbf{U}_{e,i}$ is basis function coefficient i on element e and $\phi_{e,i}$ is the i^{th} basis function on element e. Each set of basis functions has support over only a single element. We can formally consider each state $\mathbf{u}_h$ to be a member of a solution approximation space $\boldsymbol{\mathcal{V}}_h = [\mathcal{V}_h]^s$ with $\mathcal{V}_h$ defined as

$$\mathcal{V}_h = \{u \in L_2(\Omega) : u|_{\Omega_e} \in \mathcal{P}^{p_e} \; \forall \Omega_e \in \mathcal{T}_h\}, \tag{5}$$

where $\mathcal{P}^{p_e}$ is the set of polynomials of order p_e on element e. To discretize and solve the system of equations we first find the weak form of the Navier-Stokes equations. To do so we multiply Equation 3 by test functions, integrate by parts, and couple elements via numerical fluxes,

$$\begin{aligned} &\int_{\Omega_e} \mathbf{w}_h^T \frac{\partial \mathbf{u}}{\partial t} \, \mathrm{d}\Omega \; - \int_{\Omega_e} \nabla \mathbf{w}_h^T \cdot \left[\vec{\mathbf{F}}(\mathbf{u}_h) - \vec{\mathbf{G}}(\mathbf{u}_h, \nabla \mathbf{u}_h)\right] \mathrm{d}\Omega + \\ &\int_{\partial \Omega_e} \mathbf{w}_h^T \left[\widehat{\mathbf{F}}(\mathbf{u}_h^+, \mathbf{u}_h^-) - \widehat{\mathbf{G}}(\mathbf{u}_h^+, \mathbf{u}_h^-, \nabla \mathbf{u}_h^+, \nabla \mathbf{u}_h^-)\right] \cdot \vec{n} \, \mathrm{d}S \; - \int_{\partial \Omega_e} \left(\mathbf{u}_h^+ - \{\mathbf{u}_h\}\right)^T \vec{\mathbf{G}}(\mathbf{u}_h^+, \nabla \mathbf{w}_h^+) \cdot \vec{n} \, \mathrm{d}S = \mathbf{0}, \; \forall \mathbf{w}_h \in \boldsymbol{\mathcal{V}}_h. \end{aligned} \tag{6}$$

$\partial \Omega_e$ represents the element boundary, and on that boundary, $(\cdot)^+$ and $(\cdot)^-$ represent quantities taken from the current and neighboring element, respectively. Approximate numerical fluxes are denoted by $\widehat{(\cdot)}$, $\{\cdot\}$ represents an face average or boundary value, and $\vec{n}$ is the outward pointing normal vector. To create our training and testing data we use *eddy*, a DG solver [20, 21] from NASA Ames. For the numerical fluxes $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{G}}$, *eddy* uses Ismail-Roe [22, 23] and the second form of Bassi and Rebay [24], respectively.

Super-resolution, even in a fluids context, will typically use a down-sampling operator that convolves the input flow field with a down-sampling kernel

$$\mathbf{u}_H = \mathcal{M}(\mathbf{u}_h) = \mathbf{u}_h \otimes \mathcal{K}, \tag{7}$$

where $\otimes$ is a convolution operator and $\mathcal{K}$ is a blurring kernel. Our down-sampling model in DG is quite different. To down-sample a solution from a fine approximation space $\mathbf{u}_h$, we seek a solution in a coarse approximation space $\mathbf{u}_H$ that minimizes the difference between the two

$$\mathbf{u}_H = \arg\min_{\mathbf{u}_H} \int_{\Omega} |\mathbf{u}_h - \mathbf{u}_H|^2 \, \mathrm{d}\Omega. \tag{8}$$

This is a least-squares projection problem. After multiplying by test functions, integrating, and rearranging we find that our final down-sampling operation on basis function coefficients is

$$\mathbf{U}_H = \mathbf{M}_H^{-1} \mathbf{M}_h \mathbf{U}_h, \tag{9}$$

where $\mathbf{U}_h$ are the basis function coefficients of state $\mathbf{u}_h$, and $\mathbf{U}_H$ are the basis function coefficients of state $\mathbf{u}_H$. $\mathbf{M}_H$ and $\mathbf{M}_h$ are matrices containing the integral of each of the products of the $(p_H + 1)$ test functions multiplied by each of the

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.

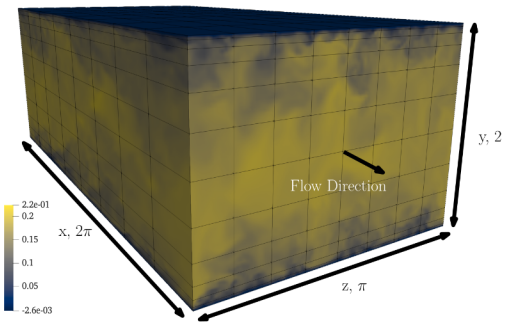


Fig. 1 x (streamwise direction) momentum contours of an original channel flow snapshot used for training and testing. High-order $p = 15$ element boundaries are shown.

$(p_H + 1)$ and $(p_h + 1)$ basis functions, respectively. This down-sampling operator returns a least-squares optimal $\mathbf{u}_H$ in the form of its basis function coefficients. After restricting the solution in this way, there is little information about the original solution remaining. For example, a $p = 3$ solution with a tensor product basis has only four data points in each direction. Additional information from neighboring elements should help alleviate this relative lack of data about the original projected state.

III. Methodology

Data for training and testing is projected from a high-resolution turbulent channel flow large eddy simulation (LES). We simulate a $Re_\tau \approx 395$ plane turbulent channel flow using *eddy*. An example snapshot is shown in Figure 1. The channel dimensions are 2π x 2 x π in the streamwise, wall-normal, and spanwise directions, respectively. The setup follows that of Moser, Kim, and Mansour [25]. A constant streamwise body force maintains turbulent flow at the specified Reynolds number. *eddy* is efficient at high-orders, and only 8 x 12 x 8 elements at $p = 15$ are required for a well-resolved LES. This simulation data could be projected directly to lower orders to form the training and testing data. Such a decision would, however, limit the size and flexibility of our tests to a single element distribution and size. For this paper we also restrict our focus to 2D elements, so some sort of projection will be required from the originally 3D elements. We have chosen to sample the channel flow at a regular grid of 512 x 256 x 512 points in the streamwise, wall-normal, and spanwise directions, respectively. We use these data to perform least-squares projection to 2D DG data at any chosen slice and mesh resolution. Our elements use tensor-product Lagrange polynomials with Chebyshev node spacing.

Our networks directly input and output DG basis function coefficients. In general, the network could use the entire flow domain to predict state on a single element, but this is not computationally feasible. Instead, we focus on a neighborhood of elements around the super-resolution target element as shown in Figure 2. We restrict our attention to only momentum coefficients in the streamwise and spanwise directions. Since our flow is nearly incompressible, the density is nearly constant at one, and the momenta are nearly identical to velocities. Testing proves that the basis function coefficients must be normalized for performance. We follow Pradhan and Duraisamy [18] for our normalization by defining the mean and root mean square (r.m.s.) values on an element for each rank as

$$\mathbf{u}_{m,e} = \frac{\int_{\Omega_e} \mathbf{u}_e \, \mathrm{d}\Omega}{|\Omega_e|} \tag{10}$$

and

$$\mathbf{u}_{\mathrm{rms},e} = \sqrt{\frac{\int_{\Omega_e} \left(\mathbf{u}_e - \mathbf{u}_{m,e}\right)^2 \, \mathrm{d}\Omega}{|\Omega_e|}} \tag{11}$$

where $\mathbf{u}_e$ is the state of the streamwise and spanwise momentum components on element e and $|\Omega_e|$ is the volume of element e. The mean and r.m.s. quantities are used to normalize each rank of the input basis function coefficients. The final network input is formed by concatenating the normalized coefficients of the central element with its neighbors

$$F_{sr}\left(\frac{\mathbf{U}_{H,c} - \mathbf{u}_{m,c}}{\mathbf{u}_{\mathrm{rms},c}}, \frac{\mathbf{U}_{H,n} - \mathbf{u}_{m,c}}{\mathbf{u}_{\mathrm{rms},c}}\right) \tag{12}$$

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.

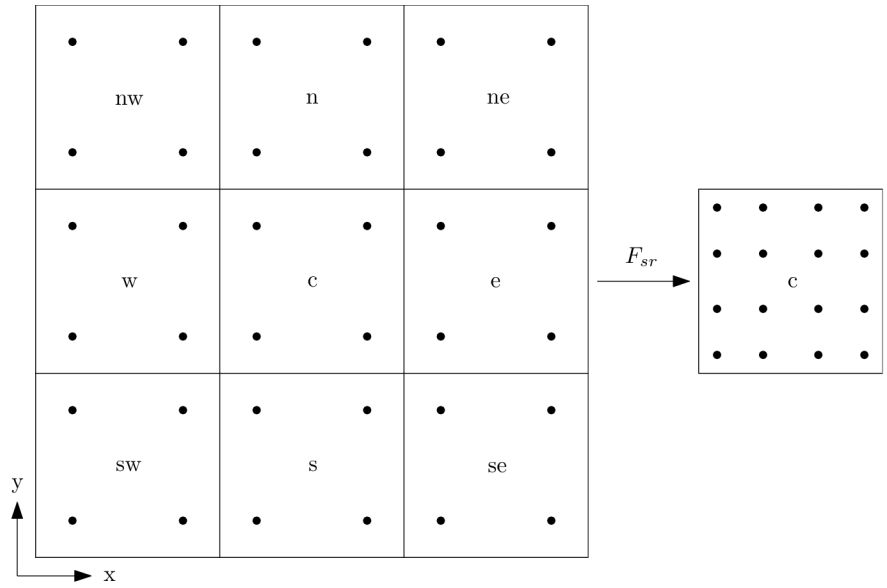


Fig. 2 $p = 1$ to $p = 3$ super-resolution of element "c" using neighboring element data. Dots indicate Lagrange node positions for each degree of freedom.

where $(\cdot)_c$ denotes central element quantities and $(\cdot)_n$ denotes quantities on all neighboring elements. Note that neighboring elements are still normalized by the central element's mean and r.m.s. values. This is because the mean and r.m.s. can vary significantly from one element to the next, so that normalizing by different values causes unnecessary discontinuities in the input data.

Our networks do not predict a final output state. Instead they predict the correction to the input central state normalized by the input r.m.s. value. This setup ensures that the network predicts only velocity differences, which has proved an effective technique for models in domains from semantic segmentation to super-resolution. The final network input and output are then

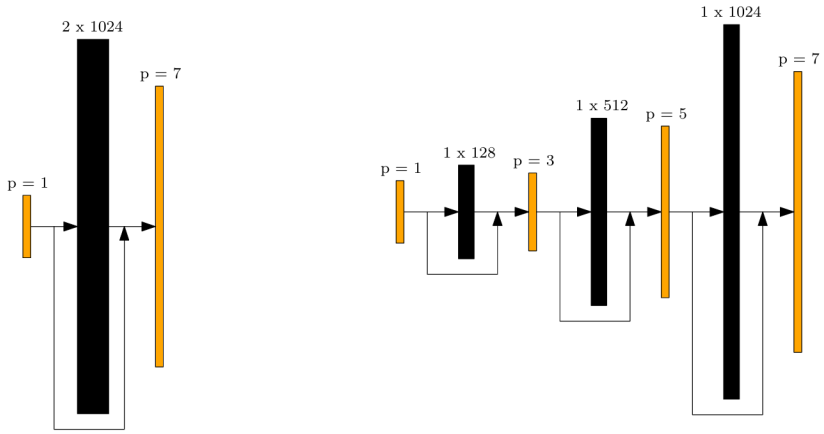
$$\mathbf{U}_{h,c} = F_{sr}\left(\frac{\mathbf{U}_{H,c} - \mathbf{u}_{m,c}}{\mathbf{u}_{\text{rms},c}}, \frac{\mathbf{U}_{H,n} - \mathbf{u}_{m,c}}{\mathbf{u}_{\text{rms},c}}\right)\mathbf{u}_{\text{rms}} + \mathbf{U}^H_{h,c} \tag{13}$$

where $\mathbf{U}^H_{h,c}$ denotes the coarse state on the central element projected into the fine space. We assume the fine space contains the coarse space making the projection operation lossless. The prolongation operation is simply Equation 9 with the fine and coarse spaces reversed with the appropriate modification to the number of test functions.

To form the F_{sr} network, the most obvious option is to construct a fully connected artificial neural network as shown in Figure 3a. We construct this network with ReLU activation functions on all layers except the last [26]. This option performs well for small jumps in polynomial reconstruction order. However, we find that for large order jumps, performance is poor. To fix this issue we propose a second network architecture where the reconstruction over large order jumps is incremental as shown in in Figure 3b. Effectively, we use several independently trained fully connected networks, each network trained for a particular order jump. Each network computes a normalized correction to the input state as described above, so each incremental network will predict increasingly finer flow features.

Each network, for both single-shot and incremental designs, is trained in a supervised framework with input / output pairs each projected from the original LES data at different orders. Training and testing data are taken from a region relatively close to the center of the channel. This ensures the character of the flow is nearly homogeneous isotropic with few wall effects. We will treat the generalization to near-wall flows in future work. The regions covered by training and testing are shown in Figure 4. While we have 256 sampled planes in the wall-normal direction, we do not use every subsequent plane for training and testing. There is an eight plane gap between each training and testing plane to ensure the data are not too similar between planes. We use the top side of the channel to gather training data and the bottom side for testing. We take care to ensure data near the center are not used for training or testing because the input data would be similar between those parts of the training and testing sets. Training data are split 80%, 20% into training and validation sets, respectively. Training uses the Adam optimizer [27], a mean squared error loss, and a learning rate of 1×10^{-4} until the validation loss stops decreasing. This usually takes on the order of 100 epochs. Example training and

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.



(a) Example fully connected single-shot network architecture to super-resolve $p = 1$ to $p = 7$.

(b) Example incremental network to super-resolve $p = 1$ to $p = 7$ with up-scaling to intermediate resolutions $p = 3$ and $p = 5$.

Fig. 3 Examples of fully connected and incremental super-resolution architectures. Black boxes represent fully connected networks labelled [hidden layers x neuron count]. Orange boxes represent input and output state at the indicated order. Direct connections from input to output represent the addition of the original state to the network output.

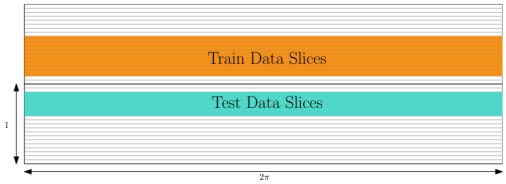
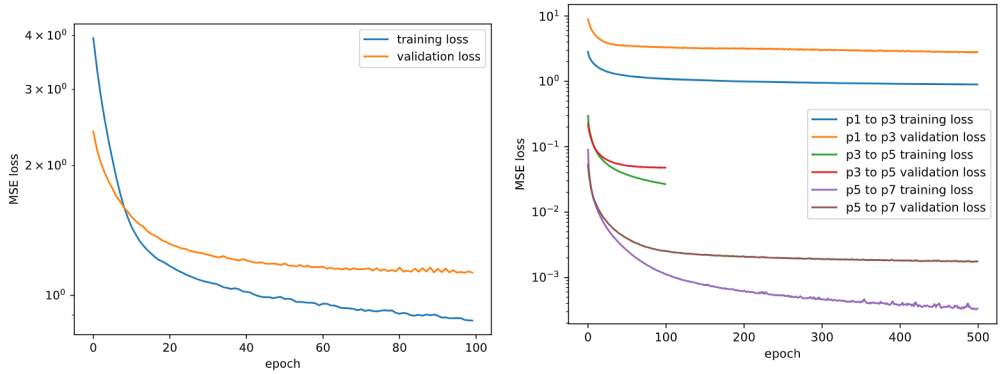


Fig. 4 Training and testing sample slice locations in turbulent channel data.

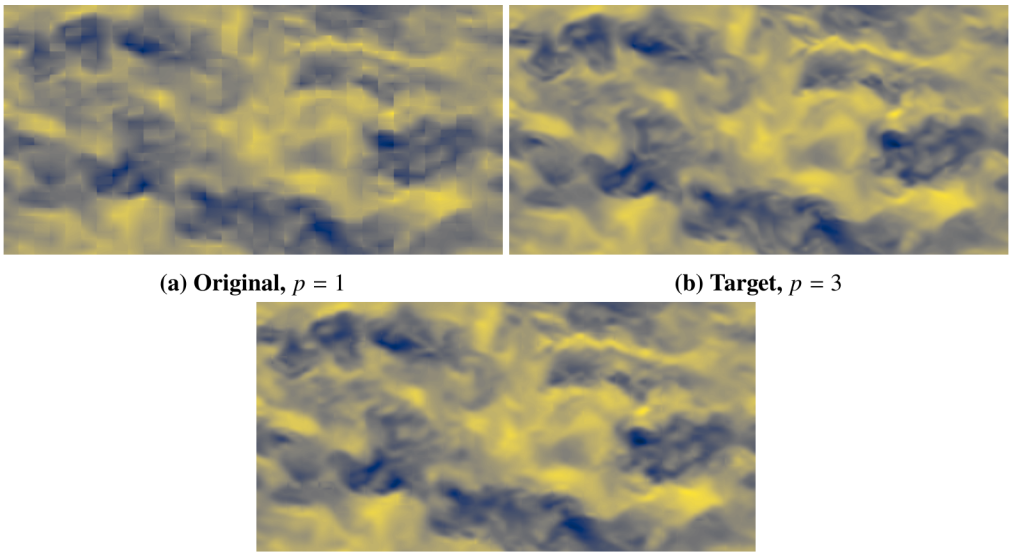


(a) History for single shot $p = 1$ to $p = 7$ network.

(b) History for each network used in $p = 1$ to $p = 7$ super-resolution.

Fig. 5 Example training histories.

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.



(a) Original, $p = 1$ **(b) Target,** $p = 3$

(c) Super-resolved, $p = 3$

Fig. 6 $p = 1$ **to** $p = 3$ **super-resolution test on 32 x 32 elements with a single hidden layer fully connected network of 128 neurons. Streamwise velocity contours at** $y^+ \approx 247$**.**

validation loss histories are shown in in Figure 5.

IV. Results

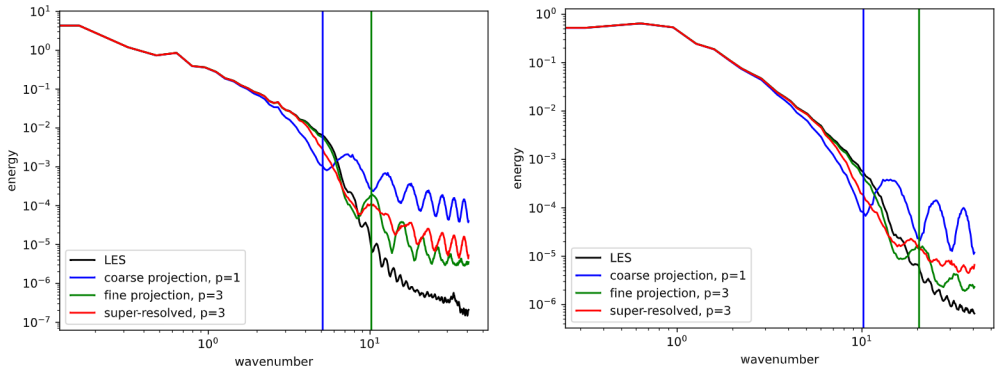
Our testing framework can choose any resolution less than or equal to the sampled LES resolution for testing. For all tests that follow, the mesh resolution, in terms of elements per direction, is chosen such that the number of degrees of freedom per direction is the same as the original simulation. This keeps the tests roughly in line with the classical notion of super-resolution, where one attempts to resolve the "true" image from a distorted or down-scaled image. Note that because we are working with DG, holding the number of degrees of freedom per direction constant is not quite sufficient to recover the original flow field. This is because high-order elements are able to represent a solution more accurately with fewer degrees of freedom and the original simulation was run at a relatively high-order $p = 15$. As a result, the target values in the following tests will deviate from the original spectral content at high-frequencies.

Our primary target in the following tests will be to recover the spectral content of the target flow field. When looking at a solution's spectral content, one will typically neglect the relatively low-energy high-frequency modes. In our situation, those high-frequency modes are beyond the frequencies representable by the polynomial solution on the elements. Instead, the high-frequency content is a result of discontinuities between elements. An accurate per-element up-scaling of a complete DG flow field should take into account each element's neighbors and not introduce spurious discontinuities between elements. To this end we seek to reduce low and high-frequency spectral errors in the following tests. To approximately distinguish between the two frequency regimes, we define a Nyquist spatial frequency $f_{nq,d}$ for each direction d as

$$f_{nq,d} = \frac{1}{2}\frac{N_{e,d}\,(p+1)}{L_d} \tag{14}$$

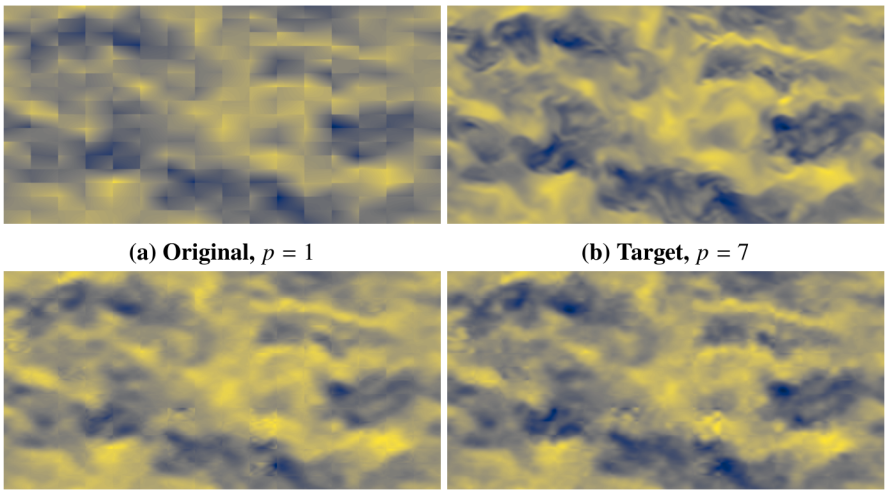
where $N_{e,d}$ is the dumber of elements in direction d, all elements are of polynomial order p, and L_d is the domain length in the d direction. This cutoff will be marked by a vertical bar in the spectral plots.

We begin by super-resolving a projected $p = 1$ field to $p = 3$. The network is a single hidden layer fully connected network with hidden layer size 128, similar to Figure 3a with different size parameters. We use 10 LES snapshots and 8 planes from each snapshot for training. Discretizing each plane into 32 x 32 elements and reserving 20% of the sample for validation leaves 65,536 training samples. Testing uses three planes from the opposite side of the channel. A spectrum is taken for each test plane on each snapshot, the spectra are then averaged over the 10 snapshots to reduce noise. The results are shown in Figures 6 and 7. The super-resolved flow field is qualitatively almost identical to the target field. The spectral content shows identical low frequencies, and nearly matching middle and high-frequency content for the



(a) Streamwise spectrum **(b) Spanwise spectrum**

Fig. 7 Streamwise and spanwise energy spectra for $p = 1$ to $p = 3$ super-resolution on 32 x 32 elements.



(a) Original, $p = 1$ **(b) Target, $p = 7$**

(c) One shot, $p = 7$ **(d) Incremental, $p = 7$**

Fig. 8 Comparison of single-shot and incremental super-resolved fields with the input and target fields. Streamwise velocity contours at $y^+ \approx 247$.

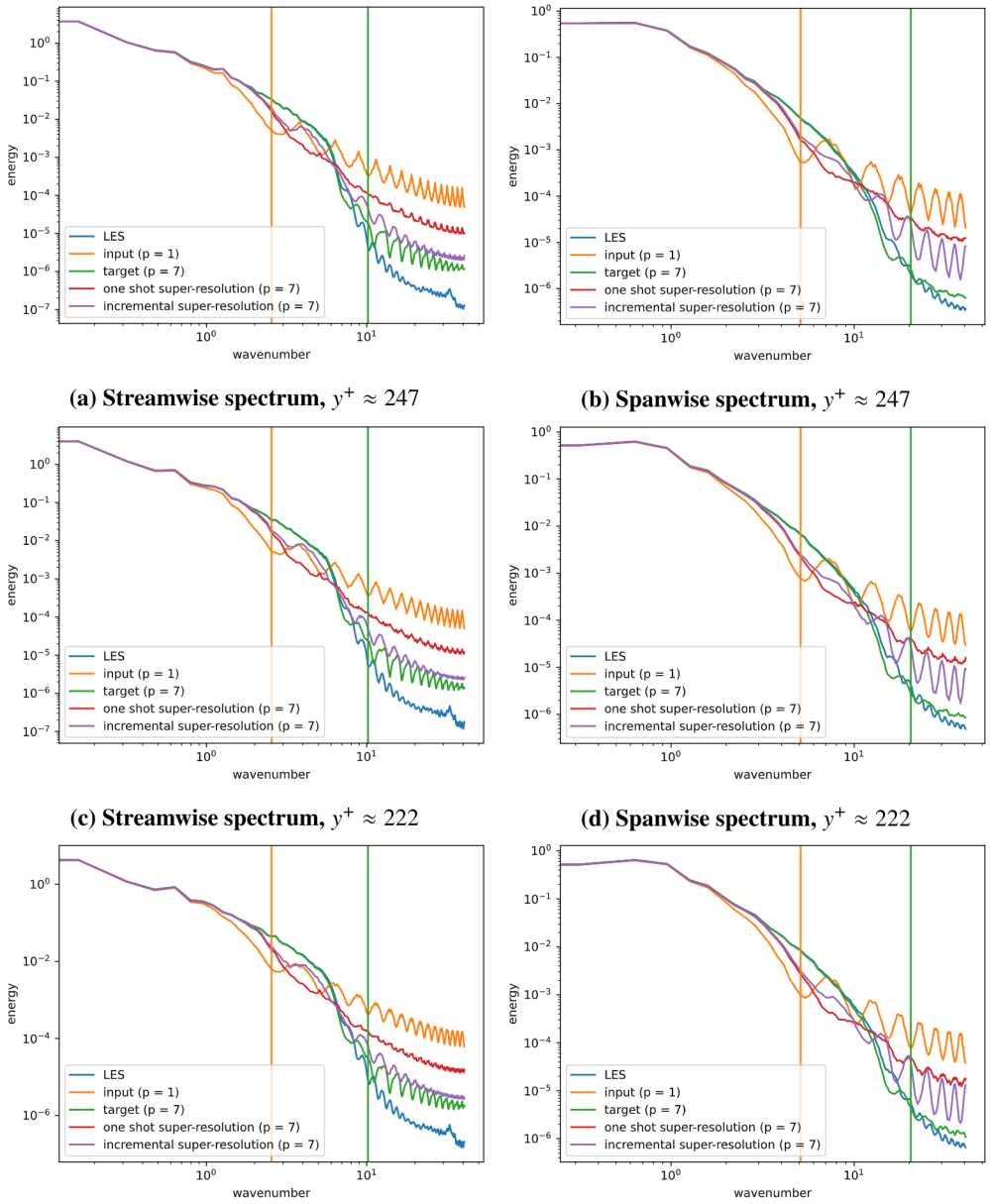
streamwise spectrum. The spanwise spectrum is somewhat less accurate but still a significant improvement over the input state.

With the apparent success of $p = 1$ to $p = 3$ super-resolution with a relatively small network, we are able to increase the difficulty. We repeat the experiment for $p = 1$ to $p = 7$ super-resolution, once again with a fully connected neural network, this time with hidden layer size 512. Once again we use the same 10 snapshots, 8 planes each for training, 3 planes for testing. Since we are using the same amount of input data with a higher target order, we have more information per sample and the number of samples decreases. Instead of projecting to 32x32 elements, we project each slice to 16x16 elements. This results in a total of 16,384 training samples, once again with 20% having been reserved for validation. With this harder problem the results have degraded significantly. Qualitatively, the resulting field lacks high-frequency content and maintains some inter-element discontinuities as shown in Figure 8c. These observations are backed up by the spectral content shown in Figure 9. Increasing the width and depth of the single-shot network fails to improve the solution in testing, indicating that the original network is sufficiently sized.

We repeat the $p = 1$ to $p = 7$ test once again, this time with the incremental super-resolution network. The training and test samples are exactly the same as in the single-shot case. The successive networks each contain a single hidden layer with sizes 128, 512, and 1024. Just as in the single-shot network case, each network is trained until its validation loss stagnates. Immediately, we see a qualitative improvement in the results of Figure 8: the flow has more high-frequency content while also reducing inter-element discontinuities. The spectral content in Figure 9 confirms the

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.



(a) Streamwise spectrum, $y^+ \approx 247$

(b) Spanwise spectrum, $y^+ \approx 247$

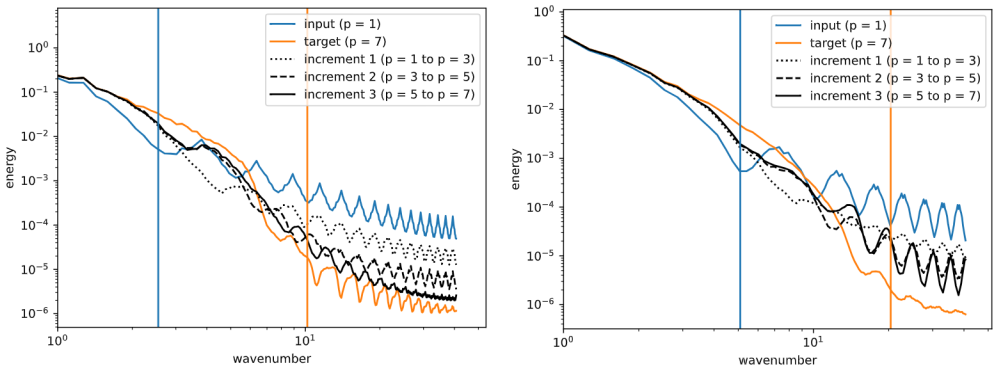
(c) Streamwise spectrum, $y^+ \approx 222$

(d) Spanwise spectrum, $y^+ \approx 222$

(e) Streamwise spectrum, $y^+ \approx 198$

(f) Spanwise spectrum, $y^+ \approx 198$

Fig. 9 Streamwise and spanwise energy spectra comparison for $p = 1$ to $p = 7$ super-resolution.



(a) Streamwise spectrum **(b) Spanwise spectrum**

Fig. 10 Spectral progression of $p = 1$ to $p = 7$ incremental super-resolution.

qualitative observations. The spectral improvement also holds across testing planes, this should be expected, since we are far from the walls of the channel, the flow should be of similar character at all test planes. The spanwise spectral content does not appear to be as improved as the streamwise spectral content. This may be due to the fact that the input data has the same number of degrees of freedom in the streamwise and spanwise directions despite the streamwise direction being twice as long. The spanwise direction is therefore able to represent higher frequencies which may be more difficult to capture.

Since the incremental super-resolution method computes increasingly high-order corrections as a state makes its way through the network, it may be of interest to see which spectral frequencies are affected by each step. To this end, in the case of $p = 1$ to $p = 7$ super-resolution with order increments of two, one would expect the lowest frequencies to be handled first, then increasingly high frequencies as the network progresses. Figure 10 shows the spectral progression through the iterations of super-resolution for the streamwise and spanwise spectra. In the streamwise case it appears that the initial hypothesis is roughly true below f_{nq} of the $p = 7$ state. It also appears that each increment reduces the level of inter-element discontinuity as shown by the decreasing energy content above f_{nq}. The spanwise spectrum is once again less impressive than its streamwise counterpart but the general character of our observations holds.

Figure 11 shows the progression of fields predicted by incremental super-resolution corresponding to the spectra in Figure 10. Each row compares the least-squares projected target state with the reconstruction at the corresponding stage. The overall spectral features shown in Figure 10 are apparent. While inter-element discontinuities are clearly noticeable in the initial condition, they are mostly removed by the second incremental reconstruction. Lower frequency content is also clearly added at each step, but some details are revealed that are masked by the averaging of the spectral plots. In the first reconstruction step, the noise added appears very similar to the truth case, though the result is clearly lacking some higher-frequency content. By the second reconstruction step, while the reconstruction remains mostly coherent, oscillations are apparent in some elements. This oscillatory behavior appears to carry through and increase in the third iteration. These oscillations add middling frequency content, but they are clearly of a different character than the turbulence in the projected snapshots.

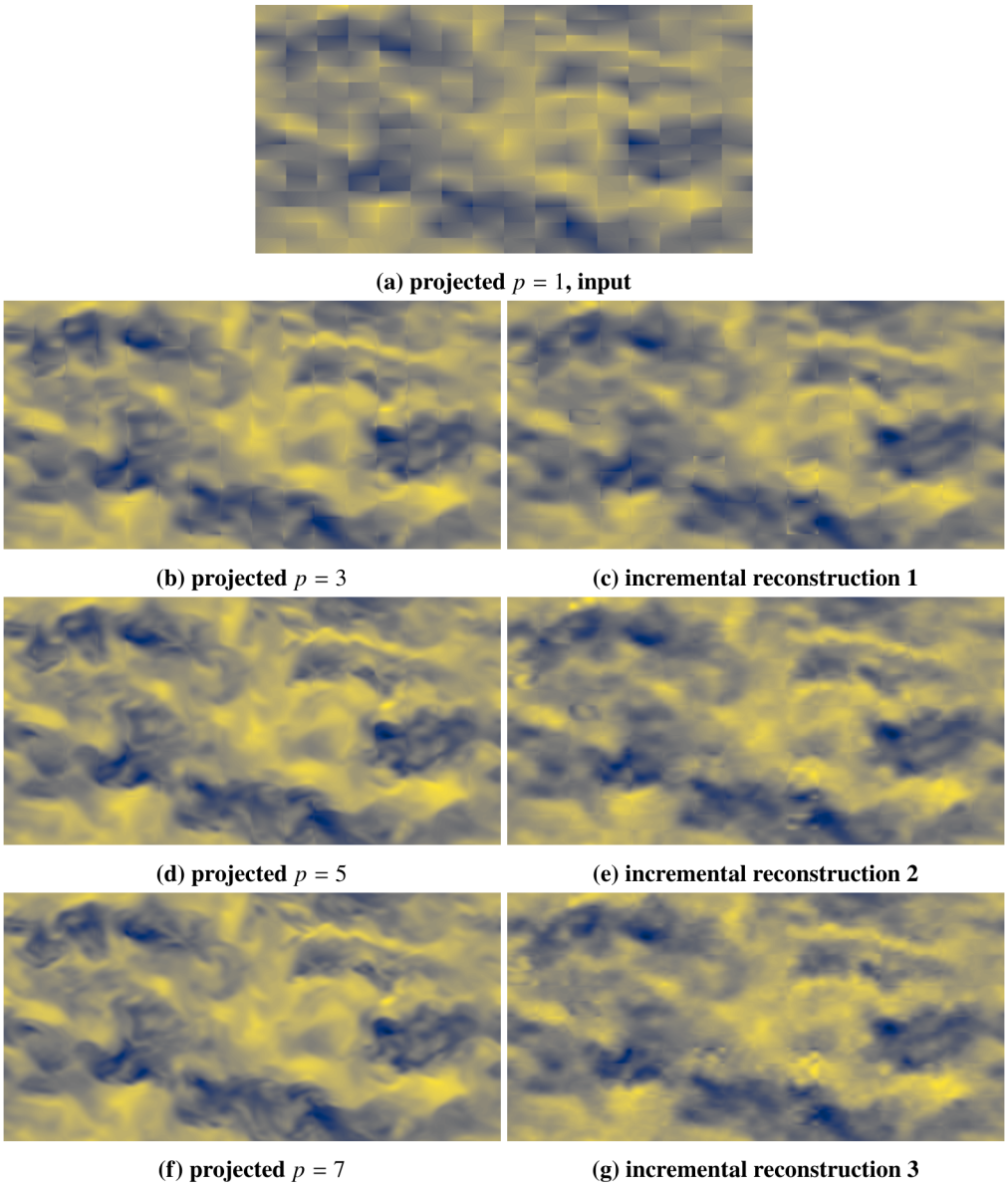
The deviation in the character of the oscillations from true projected turbulence may come down to the loss function. The loss function for all training in this paper is the mean squared error between actual and predicted basis function coefficients. Nothing in this loss encodes turbulent structures, potentially causing the spurious oscillations we see in the final super-resolved output. Also, since we use basis function coefficients as input and output, the results are dependent on the choice of basis.

V. Conclusions and Future Work

An incremental approach to super-resolution of DG elements has proven more capable than a single-shot approach at resolving low frequency content while keeping inter-element discontinuities under control. Appropriate resolution of middling frequency content has been consistently difficult in this setting even with the incremental method. This problem should not be too hard to solve since the incremental method provides some scale separation by default. It is also possible that a neural network alone is not the best solution for this type of problem, and some auxiliary frequency

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.



(a) projected $p = 1$, input

(b) projected $p = 3$ **(c) incremental reconstruction 1**

(d) projected $p = 5$ **(e) incremental reconstruction 2**

(f) projected $p = 7$ **(g) incremental reconstruction 3**

Fig. 11 Qualitative comparison of incremental super-resolution against the true solution at each reconstruction step. The left column is the truth at each incremental reconstruction step, the right column is the reconstruction. Streamwise velocity contours at $y^+ \approx 247$.

injection may be necessary. Returning these results to an adaptive framework will require the generalization of the method to different flow conditions. We intend to first extend our approach to near-wall flow regions of the channel, then to different Reynolds numbers. When these extensions work well we can generalize out technique to different element geometries and apply it in a final adaptive framework.

VI. Acknowledgments

This work is funded under the support of NASA Cooperative Agreement 80NSSC18M0149. We thank our technical monitor Gary Coleman for his ideas and advice over the course of this project.

References

[1] Farsiu, S., Robinson, D., Elad, M., and Milanfar, P., "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, Vol. 14, No. 2, 2004, pp. 47–57.

[2] Park, S. C., Park, M. K., and Kang, M. G., "Super-resolution image reconstruction: a technical overview," *IEEE signal processing magazine*, Vol. 20, No. 3, 2003, pp. 21–36.

[3] Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H., and Liao, Q., "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, Vol. 21, No. 12, 2019, pp. 3106–3121.

[4] Anwar, S., Khan, S., and Barnes, N., "A deep journey into super-resolution: A survey," *ACM Computing Surveys (CSUR)*, Vol. 53, No. 3, 2020, pp. 1–34.

[5] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, pp. 2278–2324.

[6] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D., "Backpropagation applied to handwritten zip code recognition," *Neural computation*, Vol. 1, No. 4, 1989, pp. 541–551.

[7] Dong, C., Loy, C. C., He, K., and Tang, X., "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 38, No. 2, 2015, pp. 295–307.

[8] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., "Photo-realistic single image super-resolution using a generative adversarial network," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[9] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," *Advances in neural information processing systems*, Vol. 27, 2014.

[10] Fan, Y., Shi, H., Yu, J., Liu, D., Han, W., Yu, H., Wang, Z., Wang, X., and Huang, T. S., "Balanced two-stage residual networks for image super-resolution," *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 161–168.

[11] Jiao, J., Tu, W.-C., He, S., and Lau, R. W., "Formresnet: Formatted residual learning for image restoration," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 38–46.

[12] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] Fukami, K., Fukagata, K., and Taira, K., "Super-resolution reconstruction of turbulent flows with machine learning," *Journal of Fluid Mechanics*, Vol. 870, 2019, pp. 106–120.

[14] Liu, B., Tang, J., Huang, H., and Lu, X.-Y., "Deep learning methods for super-resolution reconstruction of turbulent flows," *Physics of Fluids*, Vol. 32, No. 2, 2020.

[15] Fukami, K., Fukagata, K., and Taira, K., "Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows," *Journal of Fluid Mechanics*, Vol. 909, 2021, p. A9.

[16] Xie, Y., Franz, E., Chu, M., and Thuerey, N., "tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow," *ACM Transactions on Graphics (TOG)*, Vol. 37, No. 4, 2018, pp. 1–15.

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.

[17] Deng, Z., He, C., Liu, Y., and Kim, K. C., "Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework," *Physics of Fluids*, Vol. 31, No. 12, 2019.

[18] Pradhan, A., and Duraisamy, K., "Variational multiscale super-resolution: A data-driven approach for reconstruction and predictive modeling of unresolved physics," *International Journal for Numerical Methods in Engineering*, Vol. 124, No. 19, 2023, pp. 4339–4370.

[19] Xu, J., Pradhan, A., and Duraisamy, K., "Conditionally parameterized, discretization-aware neural networks for mesh-based modeling of physical systems," *Advances in Neural Information Processing Systems*, Vol. 34, 2021, pp. 1634–1645.

[20] Diosady, L. T., and Murman, S. M., "Design of a variational multiscale method for high reynolds number compressible flows," *21st AIAA Computational Fluid Dynamics Conference*, 2013, p. 2870.

[21] Diosady, L. T., and Murman, S. M., "Higher-order methods for compressible turbulent flows using entropy variables," *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 0294.

[22] Ismail, F., and Roe, P. L., "Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks," *Journal of Computational Physics*, Vol. 228, No. 15, 2009, pp. 5410–5436.

[23] Roe, P. L., "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of computational physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[24] Bassi, F., and Rebay, S., "GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations," *Discontinuous Galerkin Methods: Theory, Computation and Applications*, Springer, 2000, pp. 197–208.

[25] Moser, R. D., Kim, J., and Mansour, N. N., "Direct numerical simulation of turbulent channel flow up to Re τ= 590," *Physics of fluids*, Vol. 11, No. 4, 1999, pp. 943–945.

[26] Nair, V., and Hinton, G. E., "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[27] Kingma, D. P., and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Printed by [University of Michigan - 035.003.010.138 - https://arc.aiaa.org/doi/epdf/10.2514/6.2024-1983] on 14 March 2024.