Multifidelity Aerodynamic Flow Field Prediction Using Conditional Adversarial Networks

Marc Brittain*, Jethro Nagawkar†, and Leifur Leifsson‡
Iowa State University, Ames, Iowa, 50011, USA

Peng Wei§
George Washington University, Washington, D.C., 20052, USA

A novel framework for learning high-fidelity (HF) aerodynamics reconstruction from low-fidelity (LF) models is proposed to reduce the amount of HF simulations required when optimizing aerodynamic designs. The proposed framework leverages conditional generative adversarial networks (cGANs) to operate directly on images of LF outputs, generalizing our approach to arbitrary LF mesh sizes. Therefore, our approach can to be applied to more general problems, without the need to encode the underlying physics information in the neural networks. The proposed framework is then validated on a benchmark case study; flow past a backward facing step with varying step heights. Our numerical results show that our proposed framework can effectively learn the dynamics of the HF model, generating highly accurate HF model results in only 30 ms.

I. Introduction

High-fidelity (HF) simulations are often the catalyst to enable the design of many complex physical systems. These HF partial differential equation (PDE) solvers can capture the richness and non-linearities, unlike low-fidelity (LF) PDE solvers. However, the computational cost of running HF simulations can be very high resulting in long run times, quickly rendering the analysis of complex systems intractable within short time frames. Therefore, while low fidelity (LF) PDEs fail to capture the same non-linearities as in the HF PDEs, they are often the choice in early design analysis due to their quick computation time. This results in faster wall-clock designs, but the resulting design is often sub-optimal since critical design choices were made on LF simulations.

This problem becomes more prevalent during aerodynamic shape optimization (ASO). In traditional ADO methods, expensive HF simulations are used to calculate the cost function and constraint values [1–3]. These methods typically require multiple and repetitive HF model evaluations during the design process. When combined with a large number of design variables, these problems can become difficult to solve in a reasonable time period.

To reduce this computational cost, metamodeling methods have become increasingly popular [4–9]. Metamodeling methods can be classified into either data-fit methods [10, 11] or multifidelity methods [12]. Data-fit methods involve fitting a response surface through the evaluated cost function values at sampled points in the design space. Some examples of data-fit methods are Kriging [13, 14] and polynomial chaos expansions [15]. To leverage the computational advantage of LF simulations, multifidelity approaches [6, 7, 13, 16] use information from both the LF and HF simulations in an attempt to limit the number of HF simulations needed. In these approaches, the fast LF model can be used to rapidly obtain an initial approximation to the trend function, with further refinement to the true trend function with the HF model. Examples of such methods include Cokriging [17] and manifold mapping [18].

While multifidelity approaches are able to reduce the number of HF simulations, machine learning and deep neural networks have been shown to be able to solve complex PDEs, reducing the need to solve time-consuming HF CFD simulations [19–22]. In these methods, the governing physics equations can be incorporated to the loss function of the neural network which is shown to be capable of solving simple PDEs and turbulent flow problems, eliminating the need for further HF simulations. However, when approaching more complex, general problems, hand-encoding the governing physics equations into neural networks may become challenging and more time-consuming. Recently in Fukami et al. [23], the authors introduce a super resolution model that is a able to recover HF turbulent flow data

*Ph.D. Candidate, Department of Aerospace Engineering., AIAA Student Member
†Ph.D. Student, Department of Aerospace Engineering, AIAA Student Member
‡Associate Professor, Department of Aerospace Engineering, AIAA Senior Member.
§Assistant Professor, Department of Mechanical and Aerospace Engineering, AIAA Senior Member.

from LF data, without the need to encode the underlying physics equations. The framework proposed in this article similarly leverages the general adversarial networks, but instead the model operates directly on an image of the resulting LF outputs instead of the LF outputs themselves. This allows the proposed framework to be invariant to the space discretization and can handle non-uniform discretizations which is often the case in ASO.

In the deep learning community, multifidelity approaches have been extensively explored in super image super resolution (SISR), given the many recent successes with deep convolutional neural networks (CNNs) [24, 25]. In SISR, the idea is to generate a visually high-resolution output from a low-resolution input. The challenge, however, is that one low-resolution input can map to many high-resolution solutions. Therefore, there have been many different methods proposed to overcome this challenge. Traditionally, SISR has been approached from interpolation [26] and model [27] based methods. More recently, the state-of-the-art of SISR has been achieved by many CNN-based learning methods[28–39]. These methods are able to achieve visually representative 4x super resolution on input images, providing a theoretical basis for mapping LF inputs to HF outputs.

While SISR techniques are often applied to pixel up-scaling problems, image-to-image translation is a technique for learning a mapping between two images that can be of the same size [40]. This technique leverages conditional adversarial networks to allow the generator to condition on a given input image and has been shown to achieve exceptional performance in a wide range of applications [40], including single-fidelity flow-field prediction [41, 42].

In this work, a multifidelity conditional adversarial network is proposed, referred to as MFD-cGAN that operates directly on an image of the resulting LF output as opposed to the output itself. This allows our framework to handle non-uniform space-discretization which is often the case in ASO. Our framework also incorporates a conditional adversarial network to improve the performance of the learned HF mapping and does not require hand-encoding the governing physics equations. The performance of our proposed MFD-cGAN framework is evaluated on an ASO case study. The next section describes the methodology of our approach. The following section describes the case study setup along with the preliminary results. Lastly, the conclusions and future work are presented.

II. Methods

In this section, the methodology behind the proposed MFD-cGAN framework is described, including input preprocessing, the MFD-cGAN generator, and output processing.

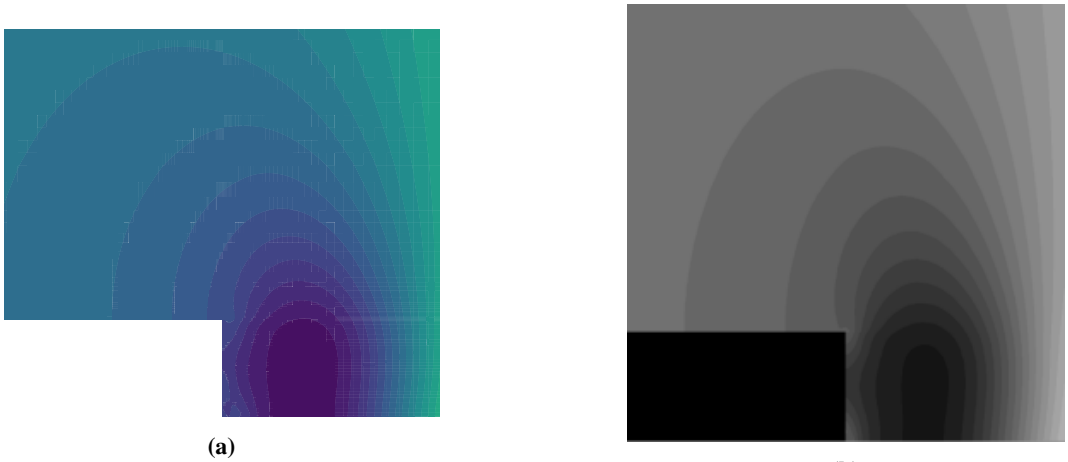
A. Input Preprocessing

In computational fluid dynamics (CFD) simulations, the mesh generated is often refined around particular regions of interest (ROI), while otherwise coarse. This is problematic as CNNs expect matrix inputs. Therefore, the first step in the MFD-cGAN framework, as illustrated in Fig. 1, is to perform preprocessing on the LF space-discretization model to convert it into a usable form for the CNN. This can be achieved by mapping the output of LF model, Y_{LF} to images in pixel space.

Images are represented by three channels (i.e., Red, Green, Blue) with varying pixel intensity values in the range of [0, 255] to produce matrices of the size $(N \times M \times 3)$, where N is the image height and M is the image width in pixels. To convert Y_{LF} to pixel space, a contour plot of the vector field produced by Y_{LF} is first generated as shown in Fig. 2(a). Then, the output of the contour plot is saved as a gray-scale image file (i.e., .png) to convert Y_{LF} to a pixel space representation as shown in Fig. 2(b).

The intuition behind this approach is further explained. By converting the contour plot to an image file, the domain is resampled by the number of specified pixels and the range of vector values is mapped to the range of pixel values ([0, 255]). In addition, by specifying the contour image as a gray-scale image, the number of image channels is reduced to one, resulting in a $(N \times M)$ representation of Y_{LF} that can be used by CNNs. The pixel representation of Y_{LF} is referred

Fig. 1 Major steps of the proposed MFD-cGAN process.



(a) **(b)**

Fig. 2 LF pressure output from a flow over a backward facing step: (a) Y_{LF} contour vector output, and (b) P_{LF} contour pixel output.

(a) **(b)**

Fig. 3 HF pressure output from a flow over a backward facing step: (a) Y_{HF} contour vector output, and (b) P_{HF} contour pixel output.

to as $P_{Y_{LF}}$.

B. MFD-cGAN Generator

In our approach, a conditional adversarial network [43] (cGAN) is introduced to train our network as described in [40]. By using this formulation, a generator network G and a discriminator network D is introduced. Given the input $P_{Y_{LF}} \in \mathbb{R}^{N \times M}$ and the true output $P_{Y_{HF}}$, the objective is to allow the generator model G learn the optimal weights θ_G, such that $G(P_{Y_{LF}}; \theta_G) = \hat{Y} \approx P_{Y_{HF}}$. In contrast, the discriminator network attempts to learn a separate set of weights, θ_D that can distinguish between $\hat{Y}$ and $P_{Y_{HF}}$. By incorporating the loss of the discriminator network into the loss of the generator network, the generator network learns how to *fool* the discriminator network, which can result in better performance as compared to using pixel-wise loss functions alone.

Both θ_G and θ_D are optimized according to their respective loss functions L_G and L_D. L_G is defined as a combination of a content and adversarial loss component as

$$L^G = L^{\text{adversarial}} + \lambda L^{\text{content}}, \tag{1}$$

where λ is a positive constant to weight the content loss component. L^{content} is often chosen to be the mean squared

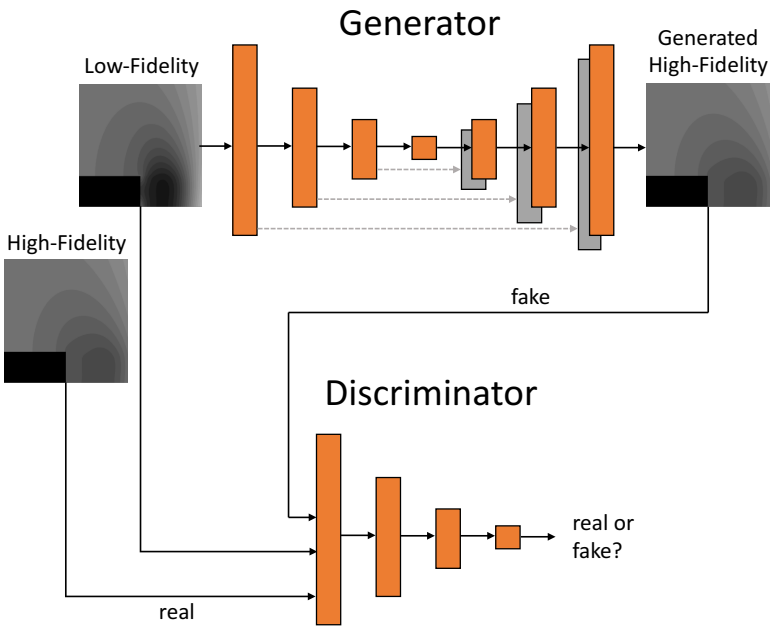


Fig. 4 Illustration of the architectures for the generator and discriminator networks.

error (MSE) or the mean absolute error (MAE). Our formulation uses the pixel-wise MAE loss that is formulated as

$$L^{\text{content}} = \arg\min_{\theta_G} |G(P_{Y_{LF}}; \theta_G) - P_{Y_{HF}}|. \tag{2}$$

We found the value of $\lambda = 100$ as recommended in [40] to provide good performance. The adversarial component is included in the total generator loss L^G to encourage the network to learn solutions that fool the discriminator network. The adversarial loss is defined as

$$L^{\text{adversarial}} = -\log(1 - D(P_{Y_{LF}}, P_{Y_{HF}}; \theta_D)) - \log D(P_{Y_{LF}}, G(P_{Y_{LF}}; \theta_G); \theta_D) \tag{3}$$

where $D(P_{Y_{LF}}, G(P_{Y_{LF}}; \theta_G); \theta_D)$ represents the probability that the generated image $G(P_{Y_{LF}}; \theta_G)$ is the true image, $P_{Y_{HF}}$, conditioned on the input image $P_{Y_{LF}}$.

The loss for the discriminator only contains the adversarial component,

$$L^D = -\log(1 - D(P_{Y_{HF}}; \theta_D)) - \log D(G(P_{Y_{LF}}; \theta_G); \theta_D). \tag{4}$$

An illustration of the generator and discriminator networks is shown in Fig. 4. The neural network architecture follows a similar setup as [40], where the generator follows a U-net architecture with skip connections, and the discriminator follows a PatchGan architecture where $N \times N$ patches of the image are classified as real or fake. In [40], the PatchGan architecture was critical for obtaining highly detailed images and in this study it was also found to lead to better performance. Each block of the generator encoding consists of a convolution layer, batch normalization, and a leaky ReLU activation. The decoding block consists of a convolution transpose, batch normalization, dropout (for first three layers), and a ReLU activation function. There is a total of 8 encoding blocks and 8 decoding blocks in the generator.

For the discriminator, a block consists of a convolution layer, batch normalization, and a Leaky ReLU activation. The output is a 30×30 patch that is used for classifying a 70×70 portion of the image. The average classification over the patch is true output of the discriminator which determines if the input image is real or fake. Both networks are optimized with the Adam[44] optimizer with a learning rate of 0.0002. Specific neural network hyperparameters follow those listed [40].

C. Output Postproccessing

The final step in the MFD-cGAN framework is to preform the output processing. In this step, a given pixel coordinate (i, j) can be mapped to a physical coordinate (x, y) by dividing the physical range by the number of pixels. For example, given a physical x-coordinate range of (x_{min}, x_{max}) and an image with a width of w pixels, the physical x coordinate can be obtained as

$$x_i = \frac{i \cdot (x_{max} - x_{min})}{w}, \quad \forall \ i \in [1, w]. \tag{5}$$

Similarly, for a given physical variable, v, the variable in the range of pixel values v^p can be mapped back to the physical variable range given the minimum and maximum variable values used during training

$$v_{i,j} = v^p_{i,j} \cdot (v_{max} - v_{min}) + v_{min}, \quad \forall \ i, j. \tag{6}$$

III. Numerical Experiments

In this section, the setup for the case study and neural network training is described, along with our numerical results.

A. Problem Formulation

The case study chosen in this work is the flow past a backward facing step. The domain for this case study is shown in Fig. 5. The inlet velocity is set to a value of 44.2 m/s, while the step height h value is 12.7 cm. The rest of the domain is scaled based on the dimensions given in Fig. 5. This setup is chosen in order to validate the CFD setup with experimental data from Driver and Seegmiller [45].

The mesh is generated using blockMesh in OpenFOAM version 5.0 [46]. Mesh $L1$ (Table 1) is shown in Fig. 6. The mesh is refined near the upper and lower walls as well as the step to ensure that the first cell thickness is less than a y^+ value of one. The grid independent results is shown in Table 1.

OpenFOAM version 5.0 [46] is used as the CFD solver for this work. The Spalart-Allmaras [47] Reynolds averaged Navier-Stokes (RANS) turbulence model is used along with the steady state simpleFoam solver to simulate the flow past the backward facing step. The boundary conditions used in this study is shown in Fig. 5. The inlet has a velocity of 44.2 m/s, while the outlet is set to a zero-gradient pressure boundary condition. The viscosity (ν) of the fluid is set to a value of 1.56×10^{-5} m^2/s. 10^{-6} is chosen as the convergence criteria for the residuals of both pressure and velocity. Table 1 gives the simulation time for each of the grids used in the mesh independence study as well as the boundary layer reattachment length (x_{RAL}) downstream of the step. From Table 1, the reattachment length moves closer to the experimental value while refining the mesh. In this study, mesh $L1$ is used as the HF mesh, while mesh $L2$ is used as the LF mesh. The step height h is varied between 10.0 cm and 1 m to train the networks.

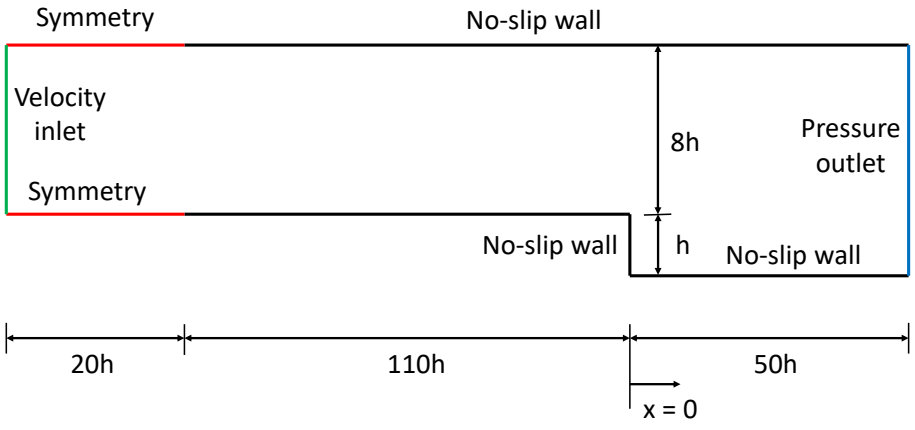


Fig. 5 Domain and boundary conditions of the backward facing step.

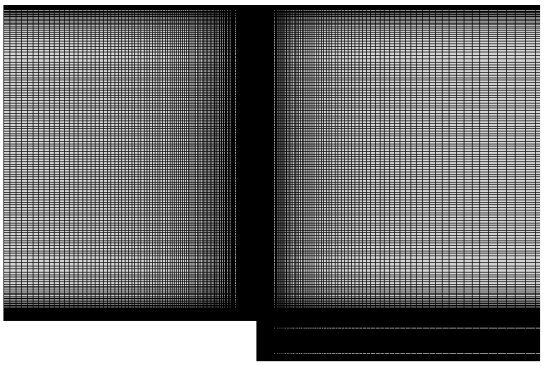


Fig. 6 Mesh $L1$ generated for the backward facing step.

Table 1 Grid convergence study for the validation case.

Mesh	No. of cells	x_{RAL}	Sim time*, s
L3	7,620	5.97h	12.5
L2	30,730	6.03h	49.3
L1	123,000	6.06h	343.4
L0	408,000	6.29h	2912.5
Exp	-	6.26h	-

*Computed on a high-performance cluster with 16 processors.

IV. Numerical Experiments

A. Network Training

To train the networks in the MFD-cGAN framework, 98 LF and HF pressure contour images were collected from the final time-step of the CFD simulation with varying step heights. The images were then randomly split into 65 training images, 10 validation images, and 23 testing images. The LF and HF images were (256×256) pixels. A similar training process as described in [40] was followed to train the generator and discriminator networks. Training completes when the content loss has not reached a new minimum in over 100 epochs, where an epoch is defined as one forward and backward pass of the network through the entire training dataset. In the training run shown, 603 training epochs were executed. As shown in Fig. 7(a), the content loss is decreasing and converging to 0, representing similarity between the generated and real pressure fields. As seen in Fig. 7(b), the total generator loss follows a similar trend as the content loss, converging to 0. The discriminator loss is shown in Fig. 7(c). For the discriminator, the loss should not approach 0 as that would represent a mode-collapse in the network and deteriorate the generator performance. The loss of the discriminator should be greater than 0 and stable to ensure that the network is accurately classifying the true images and falsely classifying the generated images, which is observed in Fig. 7(c). All of the networks were trained on an NVIDIA RTX 2080 TI GPU and a 16-core AMD Ryzen Threadripper 2950x CPU.

B. Generalization

After training, the generator was evaluated on the 23 testing images to evaluate the performance of the proposed framework when generalizing to new step heights not present in the training set. Figure 8 shows three different arbitrarily selected generated images from the testing set with their corresponding input and ground truth image for comparison. The generated image is shown on the right-most image with the step height provided as well. Interestingly, from visual inspection all three of the generated images closely resemble the pressure contour of the ground truth (HF) images. In contrast, it can be seen how the input image contours are vastly different from the ground truth, showing that the proposed framework is capable of effectively mapping between LF and HF data.

Given that splitting the data into training, validation, and testing was at uniformly selected, 5 different training seeds were run to capture the mean and standard deviation of performance across runs. Figure 9 shows the mean L1 norm

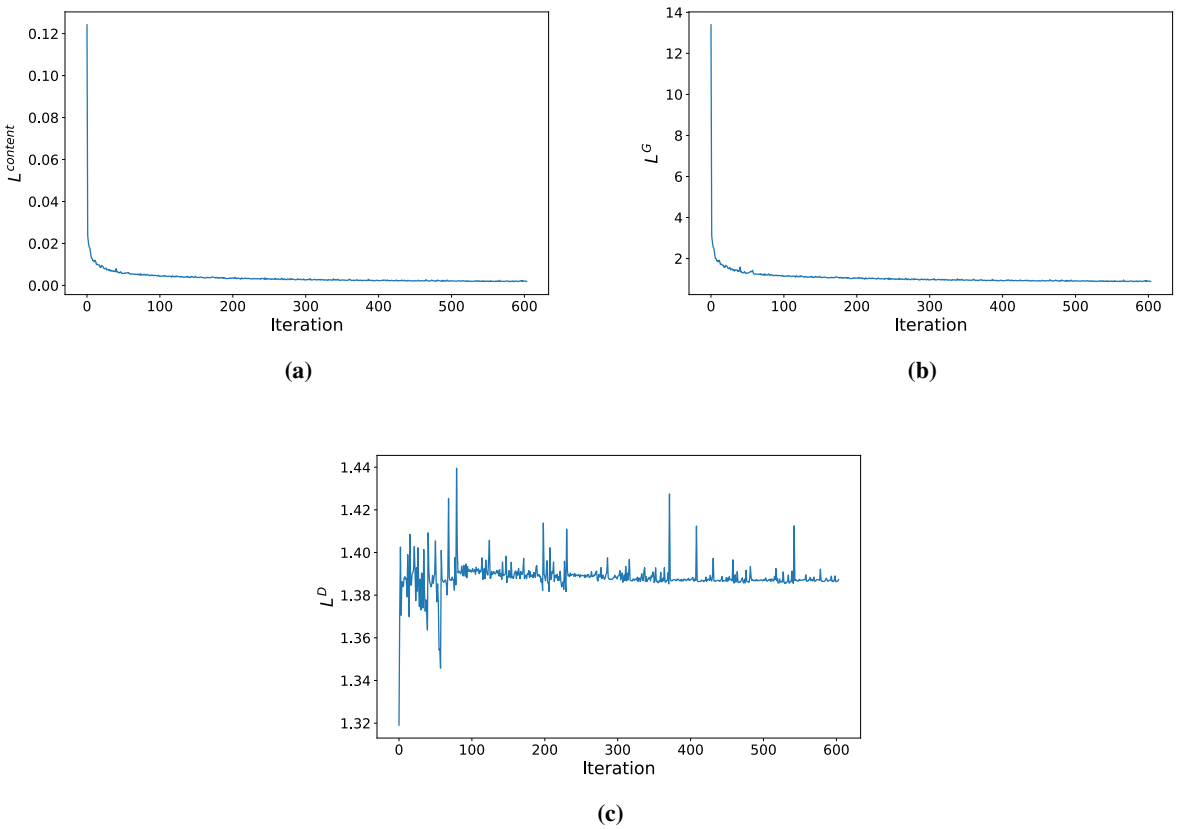
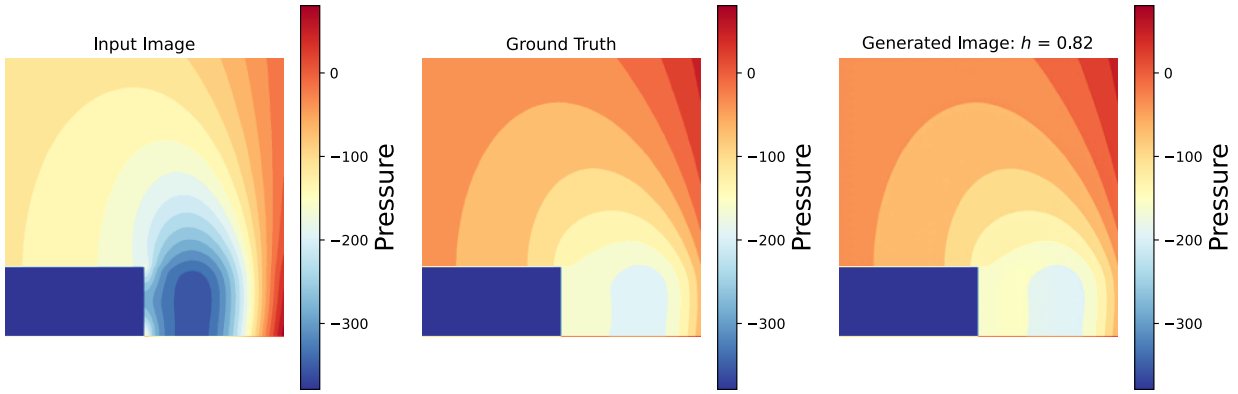


Fig. 7 Loss of the MFD-cGAN networks during training: (a) content loss (L1 norm), (b) generator loss, and (c) discriminator loss.

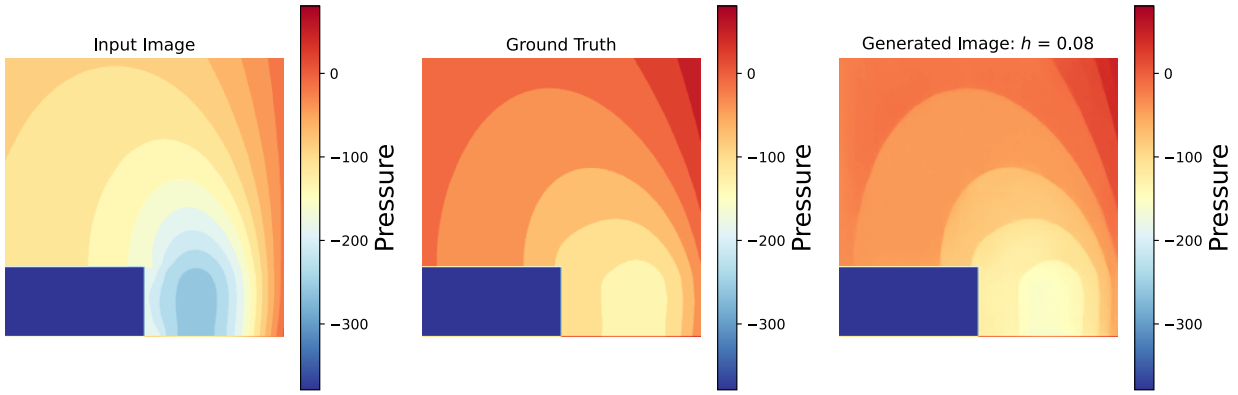
across the 5 training seeds for (a) LF and HF testing images, and (b) generated and HF testing images. The mean L1 norm is first computed over the testing images before taking a second mean over the training seeds. From Fig. 9(a), it can be seen that the L1 norm between the LF and HF testing data is very large with the maximum values around the step. In contrast, in Fig. 9(a) the L1 norm between the generated and HF testing data is very small and hardly identifiable.

In Fig. 10, the maximum average loss (average across the 5 training seeds and testing images) is recorded for each x and y pixel coordinate. This provides the worst possible error from the viewpoint of each dimension. In Fig. 10(a), it is observed that around x-coordinate 160, the maximum error is recorded for both the LF and generated image. x-coordinate 160 is directly to the right of the step. This is in agreement with the L1 error intensity shown in Fig. 9 and it can be seen how the error increases around the step location. Similarly, Fig. 10(b) shows the maximum error around y-coordinate 220 which is towards the bottom of the image, just below the step (top of the image is y = 0). The mean of L1 loss (average across the 5 training seeds, the 23 testing images, and the image x,y domain) provides a scalar comparison between the performance of LF and generated data. For the LF data, the mean of the L1 loss was 68.203 ± 0.169. For the generated data, the mean of the L1 loss was 2.782 ± 0.328, significantly outperforming the LF data.

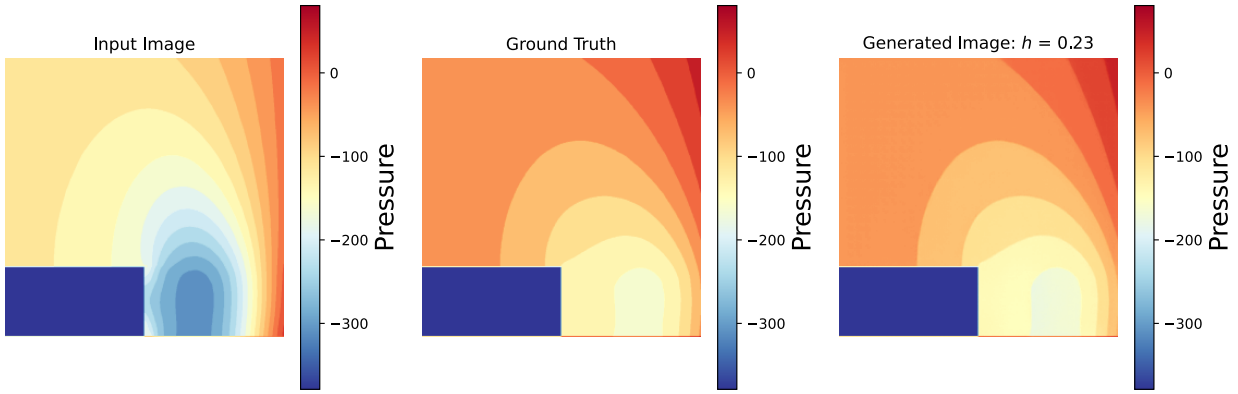
In all results shown, it can be seen that the generated data closely resembles the HF data, providing a low-cost representation of the HF data, given computationally expensive CFD simulations can be avoided. In addition, design choices made from only LF data can lead to suboptimal and slow-to-converge design optimization. Therefore, by using the generated data, more useful design choices can be made at each iterations of an ASO design loop, reducing the number of iterations to obtain an optimal design.



(a)

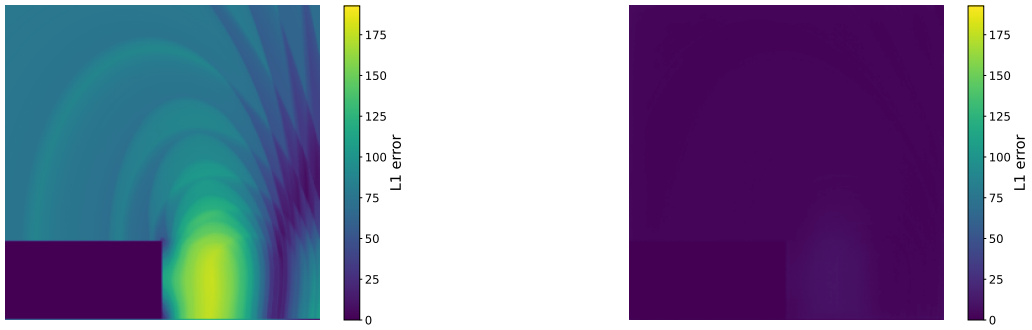


(b)



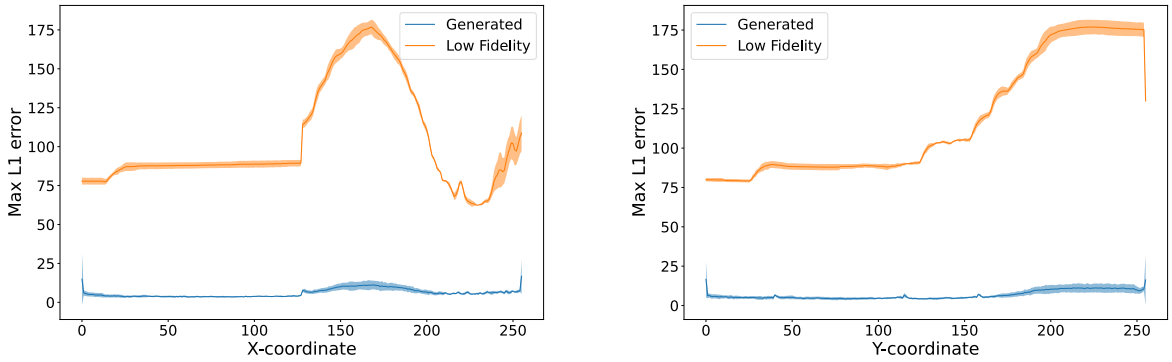
(c)

Fig. 8 Three arbitrarily selected step heights from the testing set with the corresponding LF input image, HF ground truth image, and generated HF image: (a) $h = 0.82$**, (b)** $h = 0.08$**, and (c)** $h = 0.23$**.**



(a) **(b)**

Fig. 9 L1 error heat map between: (a) LF and HF data, and (b) generated and HF data. Both images are shown in the same error scale, illustrating the similarity between the generated and HF data.



(a) **(b)**

Fig. 10 Mean and standard deviation of the Max L1 error over: (a) x-coordinate domain, and (b) y-coordinate domain for the 5 different seeds on the testing data.

C. Inference Time

Since our objective is to reduce the overall ASO problem time, the reconstruction time, or inference time of the generator to reconstruct a HF image is also evaluated. To do this, the generator is queried 1,000 times on a single LF input and record the mean inference time and standard deviation. The inference time was $30 \pm 1.43\ ms$.

V. Conclusion

In this work, a novel framework for learning high-fidelity (HF) aerodynamics reconstruction from low-fidelity (LF) models using conditional adversarial networks is proposed. Our approach is able to reduce the overall time for ASO problems by eliminating the need for running HF simulations when optimizing aerodynamic designs. According to our knowledge, this is the first investigation the feasibility and performance of using conditional adversarial networks for multifidelity ASO problems, which is shown to be able to solve complex problems. In future work, the MFD-cGAN framework will be applied to multifidelity airfoil flow field generation and integrated within ASO design optimization.

Acknowledgements

The first author is partially funded by the National Science Foundation under Award No. 1846862 and the NASA Iowa Space Grant under Award No. NNX16AL88H. The second and fourth authors acknowledge the support of the National Science Foundation under award No. 1846862.

References

[1] Skinner, S. N., and Zare-Behtash, H., "State-of-the-art in aerodynamic shape optimisation methods," *Applied Soft Computing*, Vol. 62, 2018, pp. 933–962. https://doi.org/"10.1016/j.asoc.2017.09.030".

[2] Mader, C. A., and Martins, J. R. R. A., "Derivatives for Time-Spectral Computational Fluid Dynamics Using an Automatic Differentiation Adjoint," *AIAA Journal*, Vol. 50, 2012, pp. 2809–2819.

[3] Leung, T. M., and Zingg, D. W., "Aerodynamic shape optimization of wings using a parallel newton-krylov approach," *AIAA journal*, Vol. 50, No. 3, 2012, pp. 540–550.

[4] Koziel, S., Tesfahunegn, Y., and Leifsson, L., "Variable-fidelity CFD models and co-Kriging for expedited multi-objective aerodynamic design optimization," *Engineering Computations*, Vol. 33, No. 8, 2016, pp. 2320–2338. https://doi.org/10.1108/EC-09-2015-0277.

[5] Amrit, A., Leifsson, L. T., Koziel, S., and Tesfahunegn, Y. A., *Efficient Multi-Objective Aerodynamic Optimization by Design Space Dimension Reduction and Co-Kriging*, June 2016. https://doi.org/10.2514/6.2016-3515.

[6] Du, X., Ren, J., and Leifsson, L., "Aerodynamic inverse design using multifidelity models and manifold mapping," *Aerospace Science and Technology*, Vol. 85, 2019, pp. 371–385.

[7] Amrit, A., Leifsson, L., and Koziel, S., "Fast Multi-Objective Aerodynamic Optimization Using Sequential Domain Patching and Multifidelity Models," *Journal of Aircraft*, Vol. 57, No. 3, 2020, pp. 388–398.

[8] Tesfahunegn, Y. A., Koziel, S., Gramanzini, J.-R., Hosder, S., Han, Z.-H., and Leifsson, L., "Application of Direct and Surrogate-Based Optimization to Two-Dimensional Benchmark Aerodynamic Problems: A Comparative Study," *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech Forum, Kissimmee, Florida, 5-9 January 2015.

[9] Leifsson, L. T., and Koziel, S., *Adjoint-Based Nonlinear Output Space Mapping for Accelerated Aerodynamic Shape Optimization*, January 2017. https://doi.org/10.2514/6.2017-0706.

[10] Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., "Surrogate-Based Analysis and Optimization," *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1–28. https://doi.org/10.1016/j.paerosci.2005.02.001.

[11] Queipo, N., Pintos, S., and Nava, E., "Setting Targets in Surrogate-Based Optimization," *Journal of Global Optimization*, Vol. 55, No. 4, 2013, pp. 857–875. https://doi.org/10.1007/s10898-011-9837-4.

[12] Peherstorfer, B., Willcox, K., and Gunzburger, M., "Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization," *Society for Industrial and Applied Mathematics*, Vol. 60, No. 3, 2018, pp. 550–591. https://doi.org/10.1137/16M1082469.

[13] Forrester, A. I., and Keane, A. J., "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, Vol. 45, No. 1, 2009, pp. 50 – 79. https://doi.org/10.1016/j.paerosci.2008.11.001.

[14] Krige, D. G., "A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand," *Journal of the Chemical, Metallurgical and Mining Engineering Society of South Africa*, Vol. 52, No. 6, 1951, pp. 119–139.

[15] Blatman, G., "Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis," Ph.D. thesis, Universit'e Blaise Pascal, Clermont-Ferrand, France, 2009.

[16] Karakasis, M. K., and Giannakoglou, K. C., "On the use of metamodel-assisted, multi-objective evolutionary algorithms," *Engineering Optimization*, Vol. 38, No. 8, 2006, pp. 941–957.

[17] Kennedy, C. M., and O'Hagan, A., "Predicting the Output from a Complex Computer Code When Fast Approximations are available," *Biometrika*, Vol. 87, No. 1, 2000, pp. 1–13. https://doi.org/10.1093/biomet/87.1.1.

[18] Echeverria, D., and Hemker, P., "Manifold Mapping: A Two-Level Optimization Technique," *Computing and Visualization in Science*, Vol. 11, 2008, pp. 193–206. https://doi.org/10.1007/s00791-008-0096-y.

[19] Yang, L., Meng, X., and Karniadakis, G. E., "B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data," *arXiv preprint arXiv:2003.06097*, 2020.

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

[21] Meng, X., and Karniadakis, G. E., "A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems," *Journal of Computational Physics*, Vol. 401, 2020, p. 109020.

[22] Mao, Z., Jagtap, A. D., and Karniadakis, G. E., "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 360, 2020, p. 112789.

[23] Fukami, K., Fukagata, K., and Taira, K., "Super-resolution reconstruction of turbulent flows with machine learning," *arXiv preprint arXiv:1811.11328*, 2018.

[24] He, K., Zhang, X., Ren, S., and Sun, J., "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV*, June 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

[25] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems*, Vol. 28, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Curran Associates, Inc., December 2015, pp. 91–99. URL https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[26] Lei Zhang, and Xiaolin Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Transactions on Image Processing*, Vol. 15, No. 8, August 2006, pp. 2226–2238. https://doi.org/10.1109/TIP.2006.877407.

[27] Dong, W., Zhang, L., Shi, G., and Wu, X., "Image Deblurring and Super-Resolution by Adaptive Sparse Domain Selection and Adaptive Regularization," *IEEE Transactions on Image Processing*, Vol. 20, No. 7, July 2011, pp. 1838–1857. https://doi.org/10.1109/TIP.2011.2108306.

[28] Dai, T., Cai, J., Zhang, Y., Xia, S., and Zhang, L., "Second-Order Attention Network for Single Image Super-Resolution," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA*, June 2019, pp. 11057–11066. https://doi.org/10.1109/CVPR.2019.01132.

[29] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y., "Image Super-Resolution Using Very Deep Residual Channel Attention Networks," *Computer Vision – ECCV 2018, Munich, Germany*, edited by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Springer International Publishing, Cham, September 2018, pp. 294–310.

[30] Tai, Y., Yang, J., Liu, X., and Xu, C., "MemNet: A Persistent Memory Network for Image Restoration," *2017 IEEE International Conference on Computer Vision (ICCV), Venice*, October 2017, pp. 4549–4557. https://doi.org/10.1109/ICCV.2017.486.

[31] Tai, Y., Yang, J., and Liu, X., "Image Super-Resolution via Deep Recursive Residual Network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*, June 2017, pp. 2790–2798. https://doi.org/10.1109/CVPR.2017.298.

[32] Kim, J., Lee, J. K., and Lee, K. M., "Deeply-Recursive Convolutional Network for Image Super-Resolution," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV*, June 2016, pp. 1637–1645. https://doi.org/10.1109/CVPR.2016.181.

[33] Lai, W., Huang, J., Ahuja, N., and Yang, M., "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*, June 2017, pp. 5835–5843. https://doi.org/10.1109/CVPR.2017.618.

[34] 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.

[35] Zhang, K., Gao, X., Tao, D., and Li, X., "Single image super-resolution with non-local means and steering kernel regression," *IEEE Transactions on Image Processing*, Vol. 21, No. 11, 2012, pp. 4544–4556.

[36] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*, June 2017, pp. 105–114. https://doi.org/10.1109/CVPR.2017.19.

[37] Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., and Huang, T., "Wide activation for efficient and accurate image super-resolution," *arXiv preprint arXiv:1808.08718*, 2018.

[38] Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M., "Enhanced Deep Residual Networks for Single Image Super-Resolution," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI*, June 2017, pp. 1132–1140. https://doi.org/10.1109/CVPRW.2017.151.

[39] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Loy, C. C., "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks," *Computer Vision – ECCV 2018 Workshops, Munich, Germany*, edited by L. Leal-Taixé and S. Roth, Springer International Publishing, Cham, September 2019, pp. 63–79.

[40] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A., "Image-to-Image Translation with Conditional Adversarial Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976. https://doi.org/10.1109/CVPR.2017.632.

[41] Chen, D., Gao, X., Xu, C., Chen, S., Fang, J., Wang, Z., and Wang, Z., "FlowGAN: A Conditional Generative Adversarial Network for Flow Prediction in Various Conditions," *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020, pp. 315–322. https://doi.org/10.1109/ICTAI50040.2020.00057.

[42] Achour, G., Sung, W. J., Pinon-Fischer, O. J., and Mavris, D. N., *Development of a Conditional Generative Adversarial Network for Airfoil Shape Optimization*, 2020. https://doi.org/10.2514/6.2020-2261, URL https://arc.aiaa.org/doi/abs/10.2514/6.2020-2261.

[43] Mirza, M., and Osindero, S., "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

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

[45] Driver, D. M., and Seegmiller, H. L., "Features of a reattaching turbulent shear layer in divergent channelflow," *AIAA Journal*, Vol. 23, No. 2, 1985, pp. 163–171. https://doi.org/10.2514/3.8890.

[46] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C., "A Tensorial Approach to Computational Continuum Mechanics using Object-Oriented Techniques," *Computers in Physics*, Vol. 12, No. 6, 1998, pp. 620–631.

[47] Spalart, P., and Allmaras, S., "A one-equation turbulence model for aerodynamic flows," *30th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 6-9 January 1992. https://doi.org/10.2514/6.1992-439.