

Using Physics-Informed Generative Adversarial Networks to Perform Super-Resolution for Multiphase Fluid Simulations

Matthew Li

Microsoft Search, Assistant, and Intelligence,
Microsoft Corporation,
Redmond, WA 98052
e-mail: matthewli@microsoft.com

Christopher McComb

Associate Professor
Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: ccm@cmu.edu

Computational fluid dynamics (CFD) simulations are useful in the field of engineering design as they provide deep insights on product or system performance without the need to construct and test physical prototypes. However, they can be very computationally intensive to run. Machine learning methods have been shown to reconstruct high-resolution single-phase turbulent fluid flow simulations from low-resolution inputs. This offers a potential avenue towards alleviating computational cost in iterative engineering design applications. However, little work thus far has explored the application of machine learning image super-resolution methods to multiphase fluid flow (which is important for emerging fields such as marine hydrokinetic energy conversion). In this work, we apply a modified version of the super-resolution generative adversarial network (SRGAN) model to a multiphase turbulent fluid flow problem, specifically to reconstruct fluid phase fraction at a higher resolution. Two models were created in this work, one which incorporates a physics-informed term in the loss function and one which does not, and the results are discussed and compared. We found that both models significantly outperform non-machine learning upsampling methods and can preserve a substantial amount of detail, showing the versatility of the SRGAN model for upsampling multiphase fluid simulations. However, the difference in accuracy between the two models is minimal indicating that, in the context studied here, the additional complexity of a physics-informed approach may not be justified.

[DOI: 10.1115/1.4053671]

Keywords: machine learning for engineering applications, physics-based simulations

Introduction

Computational fluid dynamics (CFD) simulations have become an important tool in engineering design and computational design synthesis [1]. By significantly reducing the number of physical tests needed to iteratively refine the design of a product or system, CFD simulations can greatly reduce the total cost incurred during the design process. However, one of the most significant limitations of CFD is the tradeoff between fidelity and computation

time. In general, CFD simulations model fluid flow by resolving the physical domain as a discrete mesh, and creating systems of differential equations (instances of the Navier–Stokes equations) which are solved or approximated using numerical computing methods [2]. To improve the fidelity of a CFD simulation, the density of the discretized mesh must be increased. However, a denser mesh also incurs significantly longer computation times forcing engineers to make a tradeoff between required fidelity and allowable computation time. Machine learning (ML) has been applied to the field of CFD to mitigate this detail/computation time tradeoff with some success. For instance, deep neural networks can be used to increase the resolution of CFD simulations through a process known as super-resolution (SR). These studies demonstrate that deep learning accelerated CFD may help to mitigate the fidelity/computational tradeoff.

We add this burgeoning field of research by demonstrating that these methods can also be applied to multiphase flow. The addition of a second fluid phase into a CFD simulation can significantly complicate the problem by adding the phase fraction as a simulation quantity that must be computed across the mesh [3]. In this paper, we demonstrate the applicability of existing image super-resolution methods to multiphase CFD simulations. Specifically, we demonstrate the feasibility of using generative adversarial network (GAN) architecture to approximate volume-of-fluids (VOFs) multiphase fluid flow results. Furthermore, we compare two GAN variations, one which incorporates a physics-informed term in the loss function and one which does not to assess the value potential value of a physics-informed approach for this application area. The remainder of the paper is organized as follows. First, we provide a background section which reviews image super-resolution, generative adversarial networks, and super-resolution applications in CFD. Next, we detail the methodology used in this work, including synthetic data generation and GAN architecture. Finally, we present results and conclusions.

Background

The project exists at the intersection of several different fields of research, and some background is provided on each in the following sections.

Image Super-Resolution. Image super-resolution is a well-researched problem in the realm of computer vision, with a plethora of applications [4], including increasing the resolution of depth maps such as those used for pose estimation [5], improving the quality of hyperspectral images such as those taken by satellites [6], and image detection for small objects [7]. The fundamental problem addressed by image super-resolution is this: Can additional fidelity and detail be inferred from a low-resolution image? More specifically, the problem addressed in this work is a case of single image super-resolution. This canonical problem can be mathematically formulated as the creation of a mapping SR, such that

$$A_\varphi = \text{SR}(A, \varphi) \quad (1)$$

where A is an $m \times n$ matrix, φ is the scaling factor, and A_φ is the resultant high-resolution matrix, with dimensions $\varphi m \times \varphi n$. An accurate mapping requires the model to create add information while constructing the resulting high-resolution. The mapping is typically learned from a dataset containing pairs of low-resolution and corresponding high-resolution.

A large variety of unique methods and architectures have been experimented with, such as deep Laplacian pyramid networks [8], dense skip connections [9], and deep residual channel attention networks [10]. The most common benchmark model in this field of research is the super-resolution convolutional neural network (SRCNN). Originally proposed by Dong et al. [11], this model uses a convolutional neural network (CNN) to create a mapping between low-resolution and high-resolution images. Rather than

Contributed by the Computers and Information Division of ASME for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received May 11, 2021; final manuscript received January 18, 2022; published online February 21, 2022. Assoc. Editor: John Michopoulos.

attempting to directly capture high-level features, SRCNN instead minimizes a lower-level pixel-wise loss. Further refinements to this model have included fast SRCNN [12], which dramatically decreased the runtime, and very deep super-resolution [13], which substantially improved accuracy through a deep neural network architecture.

Generative adversarial networks (GANs) can also be used for super-resolution, as demonstrated by the super-resolution generative adversarial network (SRGAN), created by Ledig et al. [14]. Compared to SRCNN, SRGAN can infer very high-level features, leading to much more natural and realistic-looking super-resolved images. This can be attributed to the GAN architecture's generative capabilities, as well as a powerful loss function. This increased accuracy comes at the cost of an increased number of weighting parameters in the network, which does slightly reduce computational efficiency during training and inference.

Generative Adversarial Networks. The super-resolution algorithm applied in this work is based on a GAN architecture, which consists of training two separate deep neural networks against each other [15]. One network, the generator, is responsible for generating the actual output. The other network, the discriminator, is responsible for distinguishing between the generator's output and the ground truth data. During the training process, the generator becomes better at producing outputs that resemble the ground truth, and the discriminator becomes better at discerning them. Ideally, the generator should become strong enough to fool the discriminator. The resulting generator network can then be used for inference, while the discriminator is discarded in the final model.

The versatility of the GAN architecture has led to its application to a multitude of different problems. GANs have been demonstrated to produce remarkably good results in novel tasks, such as generating human faces [16] and transforming photographs into paintings [17]. They have also proven to be useful for many practical engineering applications, such as designing airfoils [18], predicting stress distribution in structures [19], and estimating leakage parameters for liquid pipelines [20]. However, GANs can be large in size and difficult to train and are vulnerable to mode collapse and vanishing gradients [15]. Their large number of parameters also leads to slower inference, making them less suitable for time sensitive tasks [21].

Image Super-Resolution Applied to Computational Fluid Dynamics. The resolution of a CFD simulation lies in the density of cells in the mesh, which is functionally analogous to pixels in an image. Therefore, for rectilinear meshes, image super-resolution techniques can be applied directly to CFD simulations. The work of Fukami et al. [22] was one of the earlier applications of SR to CFD, showing that turbulent flows could be reconstructed using a relatively simple CNN as well as a hybrid downsampled skip-connection multi-scale model. Another work by Liu et al. [23] produced further results with other CNN architectures as well as a multiple temporal paths convolutional network. They were able to show that deep learning methods outperformed traditional upsampling methods like bicubic interpolation but were still limited in their capabilities to produce results that follow the physical constraints of fluids [23]. Recently, GANs have applied to the CFD super-resolution problem as well, namely in the work by Xie et al. [24] with a novel application in 3D smoke diffusion.

Recent work has sought to imbue neural networks with explicit knowledge of physical phenomena resulting in the creation of physics-informed neural networks [25]. Physics-informed neural networks have been shown to produce good results for some applications, such as the modeling of materials [26] and high-speed flows [27], but show limitations in their current form that are not yet well understood [27]. For instance, it has been demonstrated that physics-informed models fail to provide good solution approximations for flow in porous media when shocks are introduced [28] and that some long short-term memory (LSTM) models may not perfectly resolve turbulent conditions [29].

Two recent physics-informed SR models for single-phase CFD simulations are MeshfreeFlowNet by Jiang et al. [30] and turbulence enrichment generative adversarial network (TEGAN) by Subramaniam et al. [31]. TEGAN was specifically trained to upsample instances of the incompressible forced isotropic homogeneous turbulence problem in 3D and is constrained by the time-dependent Navier–Stokes equations and the Poisson equation [31]. This model can upsample both pressure and velocity for four total fields in 3D (velocity in the x -, y -, z -directions as well as pressure) with substantial accuracy in each field. That work also demonstrated that adding a non-zero weight to the physics loss greatly improved the results but too high of a weight may be detrimental. This result is extremely important, not only legitimizing the use of physics-based loss functions for this field but also giving some evidence for other researchers to use while tuning the hyperparameters of their own models.

Methods

Problem Description. The goal of the current work is to use image SR techniques to increase the resolution of the fluid phase fraction generated by a solver that uses the volume-of-fluids (VOFs) method of modeling two-phase incompressible turbulent fluid flow. We make use of the InterFoam solver from OPENFOAM [32], an open-source and highly performant CFD software library commonly used in both industry and research [33]. Upsampling

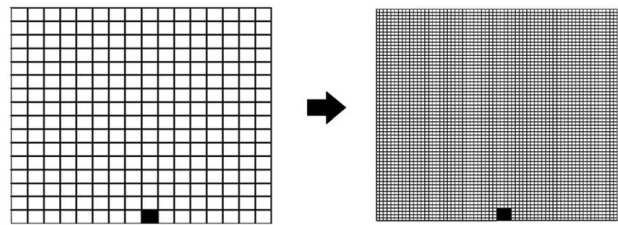


Fig. 1 Illustration of problem statement: upsampling from a 16 × 16 rectilinear mesh to a 64 × 64 rectilinear mesh

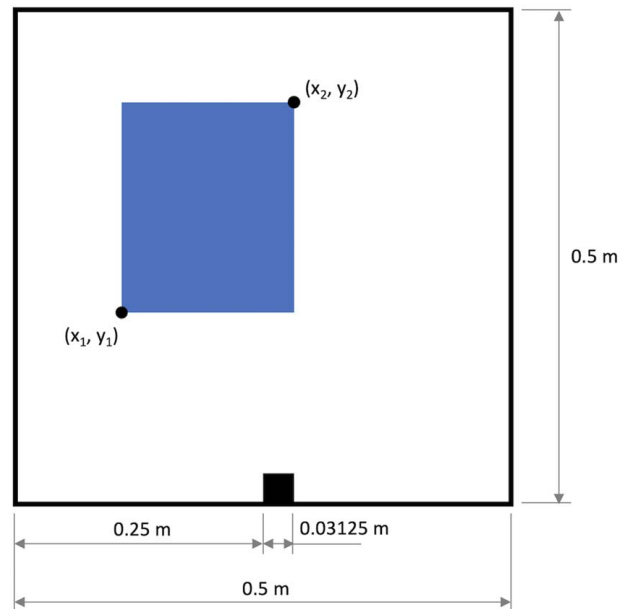


Fig. 2 Schematic diagram showing the precise dimensions used in the current study and showing dimensions of the domain and parametrization of the rectangular mass of water

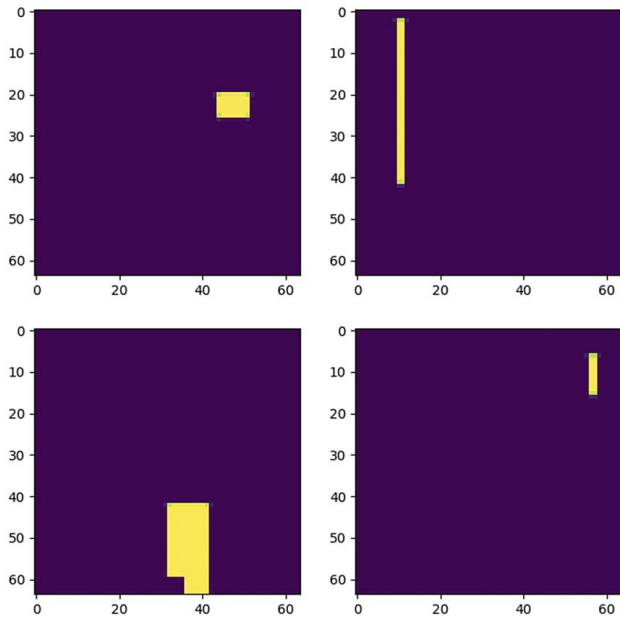


Fig. 3 Initial conditions for four different cases at time = 0 s. Lighter shades indicate cells that are predominantly filled with water, while darker shades indicate cells filled predominantly with air.

the output approximates the results of increasing the density of the discretized mesh and ideally will produce additional detail without incurring the cost of computation that comes with direct numerical solutions. To make our CFD simulations suitable for image SR models, we chose to discretize them with a uniform rectilinear mesh, which ensures the computational analogy to pixels typically used in SR. In this work, only the fluid phase fraction is considered. The model trained in this work specifically upsamples from a mesh resolution of 16×16 to 64×64 (see Fig. 1).

Training Data Generation. Training data were generated using a slightly modified version of the OPENFOAM “DamBreak” tutorial case, a two-dimensional interphase laminar flow simulation which depicts a mass of water falling from the air and onto the ground, then colliding with a solid immovable dam in the center of the floor. This creates a large amount of fluid movement for the model to learn and predict. A schematic figure of the specific dimensions used in this case is provided in Fig. 2.

A total of 800 cases were simulated with a uniform mesh density of 64×64 for 3 s with 60 discrete time-steps each (0.05 s increments), with the fluid data at each time-step serving as a data point. Every time-step constitutes a unique training sample, resulting in 48,000

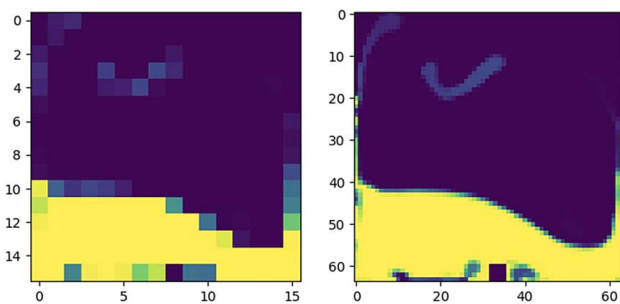


Fig. 4 Low-resolution and high-resolution dambreak cases at same time-step. Lighter shades indicate cells that are predominantly filled with water, while darker shades indicate cells filled predominantly with air.

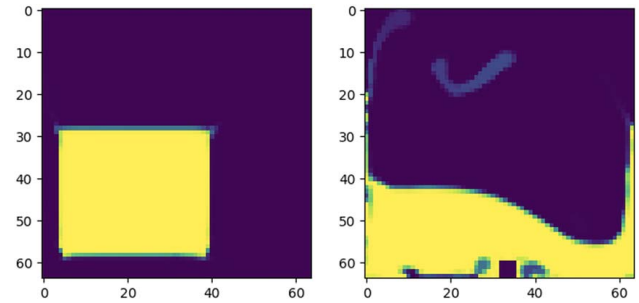


Fig. 5 Dambreak case showing before and after impact with bottom boundary. Lighter shades indicate cells that are predominantly filled with water, while darker shades indicate cells filled predominantly with air.

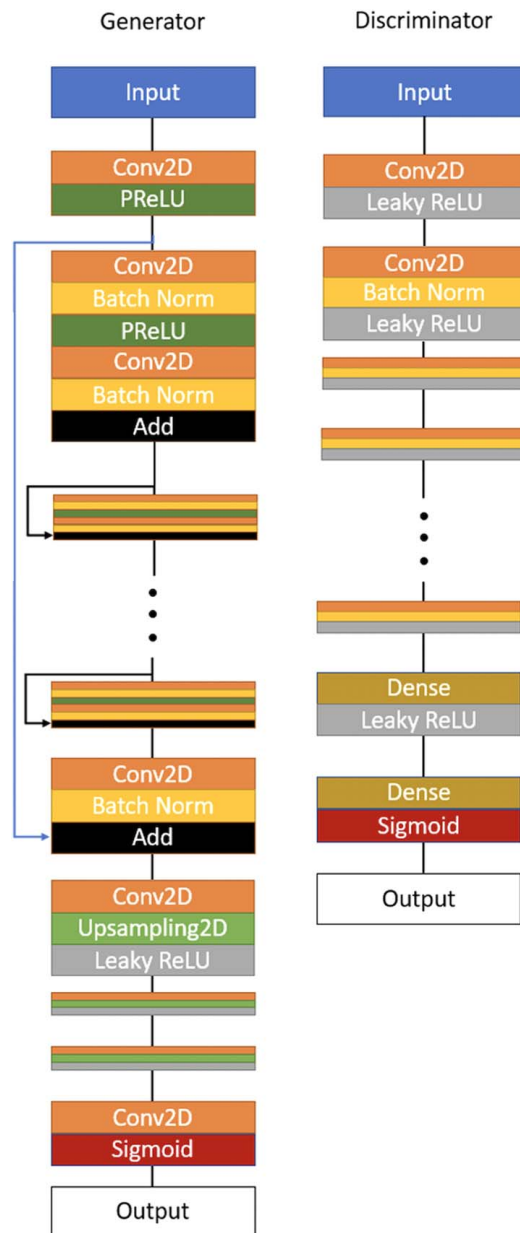


Fig. 6 Network layer diagram. Blocks without explicit labels are repeated from earlier in the respective network.

total data training samples. The initial position, size, and shape of the water mass are randomized between cases, with several examples shown in Fig. 3. The rationale for this was to create a more varied set of training samples, as changing these conditions changes the amount of kinetic energy in the system, which in turn varies the fluid behavior between cases. Specifically, the shape of the water mass was randomized in a two-step process. First, the lower left point of the mass (denoted by (x_1, y_1) in Fig. 2) was selected from a uniform random distribution across the cells in the simulation domain. Next, the upper right point of the mass (denoted by (x_2, y_2)) was selected from a uniform random distribution across the cells in the simulation domain to the upper right of (x_1, y_1) .

Each of these samples was then downsampled using linear interpolation, creating pairs of high- and low-resolution samples (see Fig. 4). This dataset was then split into two categories: pre-ground-impact and post-ground-impact (see Fig. 5). In this work, we focus on the post-ground-impact dataset since most of the fluid deformation occurs after the fluid collides with the ground. The pre-ground-impact dataset can likely be predicted with much simpler methods (i.e., projectile motion) and is not used in this paper. The model presented is trained completely on the post-ground data. This post-ground dataset has 19,664 total samples. This constitutes the portion of the 48,000 total samples that are simulated after

ground impact of the falling mass of water. Of these, 5000 samples were randomly chosen for training and testing the model.

Network Architecture. The network used here is based heavily on the SRGAN architecture, created by Ledig, and the full network layout is shown in Fig. 6. This is a GAN, and as such is composed of a generator (left) and discriminator (right).

The generator begins with a convolutional layer with a 9×9 kernel size with 64 filters and a PReLU activation, followed by 16 residual blocks with skip connections, which consist of two convolutional layers with 64 filters and a 3×3 kernel size each with a batch normalization layer, as well as two upsampling blocks which contain a convolutional layer with 256 filters and a 3×3 kernel size, and an 2D upsampling layer. Then, the last layer is a convolutional layer with one filter and a kernel size of 9×9 , with a sigmoid activation layer. The sigmoid activation function is chosen for the last layer instead of the hyperbolic tangent because the fluid phase fraction values must be between zero and one.

The discriminator consists of eight convolutional layers with a kernel size of 3×3 , split into pairs of 64 filters, 128 filters, 256 filters, and 512 filters. Each convolutional layer uses the leaky ReLU activation function. The last few layers consist of two dense layers and a sigmoid activation function.

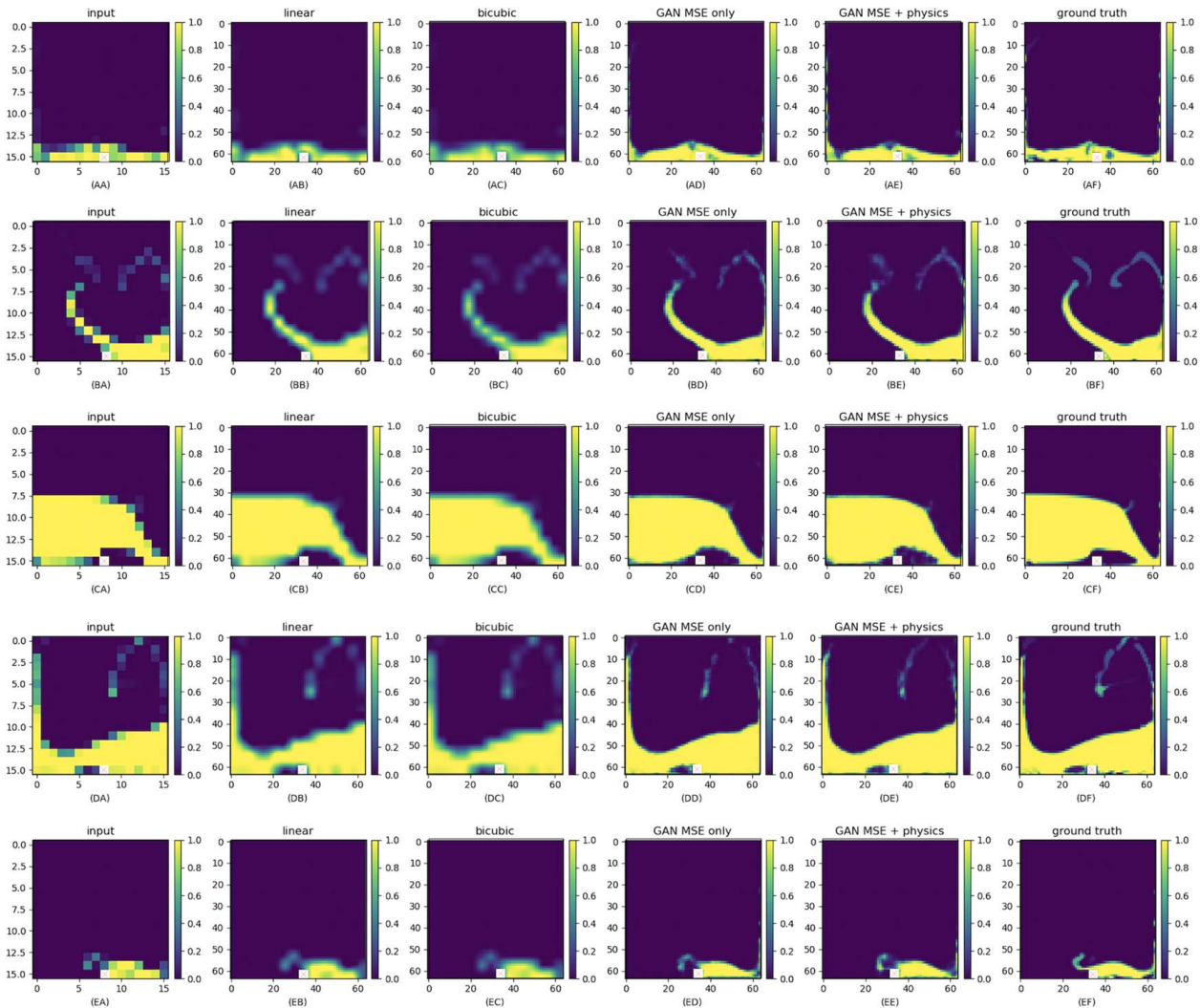


Fig. 7 Examples of model inference. From left to right, columns are the low-resolution input; high-resolution images created with linear upsampling, bicubic upsampling, the MSE-only GAN, and the MSE + physics GAN; and the ground truth high-resolution simulation. Rows in the grid represent different cases from the testing dataset, representing different time-steps and water volumes. Lighter shades indicate cells that are predominantly filled with water, while darker shades indicate cells filled predominantly with air.

Loss Function Design. While the discriminator network was compiled and trained with the original binary cross-entropy loss function, the generator was trained differently from the original SRGAN implementation. Instead, the generator was trained with a custom loss function combining a traditional mean squared error (MSE) term with additional physics-based constraints, since this dataset follows a known set of governing equations.

Physics-based loss functions for super-resolution computational fluid dynamics (SRCFD) networks have demonstrated success, as shown by Jiang [30] and Subramaniam [31]. For models that predict fluid velocity and pressure, loss functions are based on the Navier–Stokes equations, and a model that minimizes its loss should stay as consistent with these governing equations as possible, but implementing them can be difficult. However, the phase fraction follows the interphase equations, which are simpler in form. Essentially, our loss function aims to minimize the difference of the volume of liquid phase between the input and the output, in addition to minimizing pixel-wise loss. The interFoam solver models incompressible fluids, so this volume should remain consistent. The full loss function is shown in Eq. (2). The left summand is a traditional MSE term with a weighting of λ_1 and the right summand is the normalized squared difference of total water volume with a weighting of λ_2 . The variables m and n represent the height and width of the rectilinear mesh. The variables y_{ij} and \hat{y}_{ij} represent the true and predicted values of the fluid fraction inside a given voxel, respectively, while Y and \hat{Y} are matrices representing these quantities across all voxels.

$$\mathcal{L}_{\text{gen}}(Y, \hat{Y}) = \lambda_1 \cdot \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - \hat{y}_{ij})^2 + \lambda_2 \cdot \frac{1}{mn} \left(\sum_{i=1}^m \sum_{j=1}^n y_{ij} - \sum_{i=1}^m \sum_{j=1}^n \hat{y}_{ij} \right)^2 \quad (2)$$

Training and Comparison. Training was performed on a computer with an NVIDIA RTX 3070 GPU on TensorFlow 2.5.0. The network was trained using 5000 samples from the dataset, which was split into 80% for training and 20% for testing. The network was trained with a batch size of 16 for 1000 epochs, and the Adam optimizer was used with a learning rate of 1×10^{-4} .

In the current work, we train and investigate two models with different combinations of λ_1 and λ_2 (and therefore different loss functions). Namely, we investigate the case where $\lambda_1 = 1.0$ and $\lambda_2 = 0.0$, corresponding to a traditional MSE-only loss function, and the case where $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, corresponding to a novel loss function that combines a traditional term with a physics-informed term. We refer to the former as “MSE only” and to the latter as “MSE + physics.” These models are compared against traditional, non-ML upsampling approaches.

Results and Discussion

Figure 7 provides several comparisons between different upsampling approaches, namely, linear upsampling, bicubic upsampling, and the two models presented in this work. These samples were not present in the original training data. The furthest left column in the figure shows the low-resolution input, and the furthest right column shows the ground truth high-resolution simulation. The GAN models were able to capture a high degree of detail, with many high-level features preserved and visible. It seems that the generator was able to learn many of the turbulent and chaotic fluid behaviors present in the dataset. There seems to be very little difference visually between the MSE-only model and the MSE + physics model.

An examination of Fig. 7 also indicates that the GAN model outputs have much more clarity and detail than the traditional upsampling techniques. Turbulent trails of fluid were captured in the upsampled results of these models which are not present in

the linear or bicubic results. Some details that are almost imperceptible in the low-resolution input were inferred by the GAN models. For instance, thin, wispy trails of water splashing in the air that are almost completely absent from the input and are only a few pixels in size in the high-resolution simulation are captured (see Figs. 7(AE) and 7(CE)). In addition, small pockets of air resulting from the collision of the water with the bottom boundary were somewhat preserved by the model (see like in Figs. 7(BE) and 7(DE)). In some cases, like Figs. 7(BE) and 7(DE), these features are more pronounced in the MSE + physics model than in the MSE-only model. A more extensive qualitative analysis should be conducted to assess the extent to which this statement holds. These details do differ slightly from the ground truth, but the ability of the model to learn these details is still noteworthy.

Figure 8 shows the MSE loss values measured across all models with a log axis. This shows that the GAN models achieve loss that is substantially lower than nearest-neighbor, bicubic, and linear upsampling approaches.

This figure reveals quantitative evidence to back up several of the observations from Fig. 7. For instance, there is little difference between the loss values of the MSE-only and MSE + physics models. The primary different is that the MSE + physics model seems to have smaller error bounds than the MSE-only model. However, the MSE-only model does achieve some lower loss values. Both models significantly outperform bicubic and linear upsampling. The variance is significant (crossing orders of magnitude for all upsampling approaches), which may be explained by the variability of the dataset, as demonstrated in Fig. 3.

As measured by loss values only, the MSE-only model does appear to outperform the physics-informed model. It could be that MSE alone provides enough information to guide the training process on this dataset, and the scope of the output space is not further reduced by adding the physical constraint. In other words, minimizing the MSE value may already result in fluid volume consistency. More comprehensive models that reconstruct other fluid properties like pressure and velocity have a much larger output space, and therefore may benefit more from physical constraints. A more complex physics-based constraint could yield better results for the current multiphase problem.

The capabilities of the model presented here should help make the argument for further research into the field of SRCFD, because the computational resources saved could potentially be

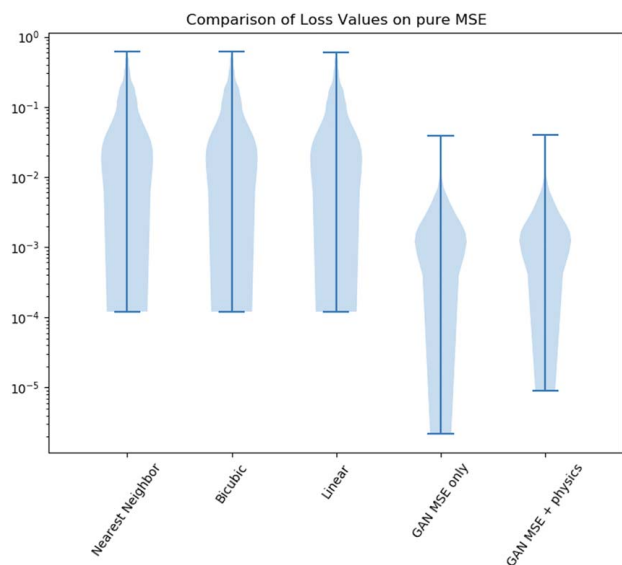


Fig. 8 Comparison of magnitude and distribution of loss values on a log scale for the different upsampling methods. These values were taken from a random sample of 10,000 data points from the dataset.

immense. Upsampling low-resolution CFD simulations to an acceptable quality using a neural network instead of directly running high-resolution CFD simulations with traditional numerical methods can save a lot of computation time, potentially leading to the streamlining of many workflows. The results presented here are not perfect, but the success found here definitely warrants further research and experimentation, nonetheless.

Conclusions and Future Work

The contributions described in this work are twofold. First, we demonstrate that existing image super-resolution methods can be applied to multiphase fluid flow with only minimal changes. Specifically SRGAN, the model demonstrated here, is capable of reconstructing turbulent multiphase flow at a higher resolution and with high accuracy than naïve upsampling approach. Second, we evaluate the potential value of a physics-informed approach to the multiphase fluid flow super-resolution task. Specifically, we show that a mass conservation term in the loss function shows little improvement over a traditional loss function.

Future work should further investigate the application of refined and modified SRCFD models to multiphase turbulent flow. CFD simulations include a time component, and subsequent time-steps are dependent on one another. Recurrent models could possibly provide improved results by taking advantage of the information contained in subsequent simulation frames. This work also focused solely on the fluid phase fraction, but a comprehensive model that reconstructs additional fluid properties like pressure and velocity for multiphase flow would potentially benefit more significantly from physics-based constraints.

The data used in this work are limited in two specific ways. First, all simulations were conducted with the same set of boundary conditions. The integration of diverse boundary conditions directly in the data representation or loss function should be explored in future work. Second, this work relied entirely on two-dimensional CFD simulations. Three-dimensional simulations incur even greater computational penalties when resolution is increased, so super-resolution should be investigated for those cases as well. This would require an updated SRGAN architecture which uses three-dimensional convolutions rather than two-dimensional convolutions. Predictions over three-dimensional domains have been demonstrated in other super-resolution tasks [34] and should be feasible for multiphase flow as well.

Acknowledgment

This material was based upon work supported by the Defense Advanced Research Projects Agency through cooperative agreement FA8750-20-C-0002. Any opinions, findings, and conclusions or recommendations expressed in this thesis are those of the authors and do not necessarily reflect the views of the sponsors.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request. The data and information that support the findings of this article are freely available.

References

[1] Bakker, A., 2006, "The Future of CFD in Engineering Design," Design Engineering Conference 16th National Manufacturing Week, Rosemont, IL, Mar. 20–23, pp. 1–25.

[2] Bhaskaran, R., and Collins, L., 2002, "Introduction to CFD Basics," Sibley School of Mechanical and Aerospace Engineering, Cornell University, pp. 1–21.

[3] Lane, G. L., Schwarz, M. P., and Evans, G. M., 2005, "Numerical Modelling of Gas-Liquid Flow in Stirred Tanks," *Chem. Eng. Sci.*, **60**(8–9), pp. 2203–2214.

[4] Wang, Z., Chen, J., and Hoi, S. C. H., 2021, "Deep Learning for Image Super-Resolution: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, **43**(10), pp. 3365–3387.

[5] Hui, T.-W., Loy, C. C., and Tang, X., 2016, "Depth Map Super-Resolution by Deep Multi-Scale Guidance," European Conference on Computer Vision, Amsterdam, The Netherlands, Oct. 11–14, Springer, Cham, pp. 353–369.

[6] Masi, G., Cozzolino, D., Verdoliva, L., and Scarpa, G., 2016, "Pansharpening by Convolutional Neural Networks," *Remote Sens.*, **8**(7), p. 594.

[7] Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., and Yan, S., 2017, "Perceptual Generative Adversarial Networks for Small Object Detection," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 21–26, IEEE, pp. 1951–1959.

[8] Lai, W.-S., Huang, J.-B., Ahuja, N., and Yang, M.-H., 2017, "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 21–26, IEEE, pp. 5835–5843.

[9] Tong, T., Li, G., Liu, X., and Gao, Q., 2017, "Image Super-Resolution Using Dense Skip Connections," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, Oct. 22–29, IEEE, pp. 4809–4817.

[10] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y., 2018, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks," Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, Sept. 8–14, pp. 286–301.

[11] Dong, C., Loy, C. C., He, K., and Tang, X., 2016, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, **38**(2), pp. 295–307.

[12] Dong, C., Loy, C. C., and Tang, X., 2016, *Accelerating the Super-Resolution Convolutional Neural Network*, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).

[13] Kim, J., Lee, J. K., and Lee, K. M., 2016, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, June 27–30, pp. 1646–1654.

[14] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., et al., 2017, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," Proceedings—30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, July 21–26, pp. 4681–4690.

[15] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014, "Generative Adversarial Nets," *Adv. Neural Inf. Process. Syst.*

[16] Wang, J., and Perez, L., 2017, "The Effectiveness of Data Augmentation in Image Classification Using Deep Learning," *Convolutional Neural Networks for Visual Recognition*.

[17] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A., 2017, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, Oct. 22–29, IEEE, pp. 2242–2251.

[18] Wang, Y., Shimada, K., and Farimani, A. B., 2021, "Airfoil GAN: Encoding and Synthesizing Airfoils for Aerodynamic-Aware Shape Optimization," *arXiv preprint*

[19] Jiang, H., Nie, Z., Yeo, R., Farimani, A. B., and Kara, L. B., 2021, "StressGAN: A Generative Deep Learning Model for Two-Dimensional Stress Distribution Prediction," *ASME J. Appl. Mech.*, **88**(5), p. 051005.

[20] Zheng, J., 2020, "A Method of Leakage Parameters Estimation for Liquid Pipelines Based on Conditional Generative Adversarial Network," International Pipeline Conference (Vol. 84461), American Society of Mechanical Engineers, p. V003T04A007.

[21] Pant, P., and Farimani, A. B., 2020, "Deep Learning for Efficient Reconstruction of High-Resolution Turbulent DNS Data," *arXiv preprint*

[22] Fukami, K., Fukagata, K., and Taira, K., 2019, "Super-Resolution Reconstruction of Turbulent Flows With Machine Learning," *J. Fluid Mech.*, **870**, pp. 106–120.

[23] Liu, B., Tang, J., Huang, H., and Lu, X. Y., 2020, "Deep Learning Methods for Super-Resolution Reconstruction of Turbulent Flows," *Phys. Fluids*, **32**(2), p. 025105.

[24] Xie, Y., Franz, E., Chu, M., and Thuerey, N., 2018, "TempoGAN: A Temporally Coherent, Volumetric GAN for Super-Resolution Fluid Flow," *ACM Trans. Graph.*, **37**(4), pp. 1–15.

[25] Raissi, M., Perdikaris, P., and Karniadakis, G. E., 2019, "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," *J. Comput. Phys.*, **378**, pp. 686–707.

[26] Liu, D., and Wang, Y., 2019, "Multi-Fidelity Physics-Constrained Neural Network and Its Application in Materials Modeling," *ASME J. Mech. Des.*, **141**(12), p. 121403.

[27] Mao, Z., Jagtap, A. D., and Karniadakis, G. E., 2020, "Physics-Informed Neural Networks for High-Speed Flows," *Comput. Methods Appl. Mech. Eng.*, **360**, p. 112789.

- [28] Wang, S., Teng, Y., and Perdikaris, P., 2021, "Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks," *SIAM J. Sci. Comput.*, **43**(5), pp. A3055–A3081.
- [29] Fukami, K., Nabae, Y., Kawai, K., and Fukagata, K., 2019, "Synthetic Turbulent Inflow Generator Using Machine Learning," *Phys. Rev. Fluids*, **4**(6), p. 064603.
- [30] Jiang, C. I., Esmailzadeh, S., Azizzadenesheli, K., Kashinath, K., Mustafa, M., Tchelepi, H. A., Marcus, P., Prabhat, M., and Anandkumar, A., 2020, "MESHFREEFLOWNET: A Physics-Constrained Deep Continuous Space-Time Super-Resolution Framework," SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, Virtual Event, Nov. 9–19, IEEE, pp. 1–15.
- [31] Subramaniam, A., Wong, M. L., Borker, R. D., Nimmagadda, S., and Lele, S. K., 2020, "Turbulence Enrichment Using Physics-Informed Generative Adversarial Networks," *arXiv preprint*.
- [32] Jasak, H., 2009, "OpenFOAM: Open Source CFD in Research and Industry," *Int. J. Naval Archit. Ocean Eng.*, **2**(1), pp. 89–94.
- [33] Deshpande, S. S., Anumolu, L., and Trujillo, M. F., 2012, "Evaluating the Performance of the Two-Phase Flow Solver InterFoam," *Comput. Sci. Discov.*, **5**(1), p. 014016.
- [34] Xu, W., Luo, W., Wang, Y., and You, Y., 2020, "Data-Driven Three-Dimensional Super-Resolution Imaging of a Turbulent Jet Flame Using a Generative Adversarial Network," *Appl. Opt.*, **59**(19), p. 5729.