Available online at www.sciencedirect.com

ScienceDirect

Proceedings of the Combustion Institute 38 (2021) 2617–2625

Proceedings of the Combustion Institute

www.elsevier.com/locate/proci

Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows

Mathis Bode[a,*], Michael Gauding[b], Zeyu Lian[a], Dominik Denker[a], Marco Davidovic[a], Konstantin Kleinheinz[a], Jenia Jitsev[c,1], Heinz Pitsch[a,1]

[a] *Institute for Combustion Technology (ITV), RWTH Aachen University, Templergraben 64, Aachen 52056, Germany*
[b] *CORIA, CNRS UMR 6614, Saint Etienne du Rouvray 76801, France*
[c] *Jülich Supercomputing Centre (JSC), Institute for Advanced Simulation (IAS), Forschungszentrum Jülich (FZJ), Wilhelm-Johnen-Straße, Jülich 52425, Germany*

Received 8 November 2019; accepted 4 June 2020
Available online 25 January 2021

Abstract

Turbulence is still one of the main challenges in accurate prediction of reactive flows. Therefore, the development of new turbulence closures that can be applied to combustion problems is essential. Over the last few years, data-driven modeling has become popular in many fields as large, often extensively labeled datasets are now available and training of large neural networks has become possible on graphics processing units (GPUs) that speed up the learning process tremendously. However, the successful application of deep neural networks in fluid dynamics, such as in subfilter modeling in the context of large-eddy simulations (LESs), is still challenging. Reasons for this are the large number of degrees of freedom in natural flows, high requirements of accuracy and error robustness, and open questions, for example, regarding the generalization capability of trained neural networks in such high-dimensional, physics-constrained scenarios. This work presents a novel subfilter modeling approach based on a generative adversarial network (GAN), which is trained with unsupervised deep learning (DL) using adversarial and physics-informed losses. A two-step training method is employed to improve the generalization capability, especially extrapolation, of the network. The novel approach gives good results in a priori and a posteriori tests with decaying turbulence including turbulent mixing, and the importance of the physics-informed continuity loss term is demonstrated. The applicability

[*] Corresponding author.
E-mail address: m.bode@itv.rwth-aachen.de (M. Bode).
[1] These authors contribute equally.

https://doi.org/10.1016/j.proci.2020.06.022

of the network in complex combustion scenarios is furthermore discussed by employing it in reactive and inert LESs of the Spray A case defined by the Engine Combustion Network (ECN).

Keywords: Generative adversarial networks; Physics-informed neural networks; Large-eddy simulation; Turbulence; ECN spray A

1. Introduction

Machine learning (ML) and deep learning (DL) have gained widespread use and have impacted many research communities and industries. The availability of exceptionally large, often extensively labeled datasets and the possibility to train large networks on graphics processing units (GPUs), reducing the training time tremendously, are two reasons for this success. Prominent applications of DL include image processing [1–3], speech recognition [4], or learning of optimal complex control [5]. These data-driven approaches have also been applied to fluid dynamics problems [6–10], including works on subfilter modeling for large-eddy simulation (LES) [11–13] based on direct numerical simulation (DNS) data. Recently, the idea of physics-informed networks [14] has emerged, where architecture or loss functions are designed to support known properties of underlying physical problems.

Neural networks have also been applied successfully to reactive flows. Some examples are the adaptive reduction scheme for modeling reactive flows by Banerjee et al. [15], artificial neural network (ANN)-based storage of flamelet solutions [16,17], and direct mapping of LES resolved scales to filtered-flame generated manifolds using customized convolutional neural networks (CNNs), as shown by Seltz et al. [18]. Additionally, regularized deconvolution methods, such as those published by Wang and Ihme [19], are closely related ideas.

Often, the applications regarding flow data are limited owing to the use of either simple networks or small, artificial datasets. Thus, many questions still remain open, such as the determination of proper network architectures for flow problems, search for hyperparameters, or improvement of the generalization ability of networks.

This work introduces the application of generative adversarial networks (GANs) [20] for subfilter modeling of turbulent flows, as GANs seems to be a flexible tool that are also promising for reactive turbulent flow simulations. GANs belong to a particular class of generative models that aim to estimate the unknown probability density underlying any observed data. The specific characteristic of this model class is the ability to perform such estimation without an explicitly provided data likelihood function. The learning takes place via an implicit generative model and only requires access to data samples from the unknown distribution. Thus, GANs perform unsupervised learning of unknown data probability distributions and do not require any labels that are necessary in supervised learning scenarios. Simply put, the particularly interesting feature of GANs is that besides a resulting generator network for modeling, a second network, the discriminator, is also used. While the generator creates new modeled data, the discriminator tries to assess if the data is real or generated and provides feedback to train the generator. Thereby, the discriminator learns better discrimination of real and modeled data, which helps the generator to generate more accurate modeled data. Precisely, estimating an unknown data probability distribution by GAN learning can be understood as a minimax zero-sum game carried out by two players, the generator and the discriminator, that are both deep networks constituting a full GAN. In this game, a generator creates samples to present them to the discriminator, while the discriminator, being confronted with a mix of generated and real data samples, has the task of guessing whether a presented input is generated or real data. So, the generator attempts to “fool” the discriminator, while the discriminator strives to better differentiate generated samples from real samples. It was shown that finding the equilibrium of this game corresponds to minimizing different distance measures between the generator model and the true data distribution [20], such as Kullback-Leibler (KL), Jensen-Shannon (JS) divergence, or Wasserstein distance, depending on a particular form of loss termed as adversarial loss.

Here, a physics-informed enhanced super-resolution GAN (PIESRGAN) is employed, built upon enhanced super-resolution GAN (ESRGAN) [2] architecture, which has been recently developed in the context of super-resolving GANs (SRGANs) [21], to reconstruct fully resolved turbulence fields from filtered data, such as from LES. To this end, the ESRGAN is extended for three-dimensional (3-D) data handling and, most importantly, endowed with physics-informed loss. Once the fully resolved data is reconstructed, a filter kernel is applied to close the filtered equations of the LES. In this work, Section 2 describes the PIESRGAN in detail and explains the network's key features that are required for an accurate reconstruction. It contains both a priori and a posteriori tests with decaying turbulence data including turbulent mixing of a passive scalar, which could represent the mixture fraction in any combustion model. Furthermore, an approach to improve the training and generalization capability, especially for extrapolation, of the trained neural network by combining fully resolved and underresolved data is discussed. In Section 3, the potential of the novel method is demonstrated by using PIESRGAN as a subfilter model for the filtered momentum and scalar equations in an LES of the Spray A case defined by the Engine Combustion Network (ECN) [22], which is a complex reactive turbulent flow featuring high Reynolds numbers and spray. The paper finishes with conclusions and recommendations for future work.

2. Modeling

A subfilter model needs to predict the subfilter statistics of fully resolved data (e. g., DNS data; denoted with "H"), knowing only the corresponding filtered data with reduced information content (e. g., LES data; denoted with "F"). Here, the fully resolved data Φ_H and the filtered data Φ_F are connected by a filter operation $\Phi_\mathrm{F} = \mathscr{F}(\Phi_\mathrm{H})$, for example, with a Gaussian filter kernel. The filtered equations, which are solved in LES, could be closed if the fully resolved data is reconstructed with an inverse filter operation $\Phi_\mathrm{H} = \mathscr{F}^{-1}(\Phi_\mathrm{F})$ that statistically restores the original fully resolved data [13].

The described challenges of subfilter models are similar to challenges faced in super-resolution imaging. Here, SRGANs have been found to be a successful tool for approximating the inverse deconvolution operator $\Phi_\mathrm{R} = \tilde{\mathscr{F}}^{-1}(\Phi_\mathrm{F}) \approx \mathscr{F}^{-1}(\Phi_\mathrm{F})$ [2,21], where Φ_R denotes the reconstructed, high-resolution data. Thus, PIESRGAN is used as approximation $\tilde{\mathscr{F}}^{-1}$. For example, if ϕ_F^n denotes a discretized filtered solution at time step n, the resulting simulation workflow for closing unclosed terms ψ_F^n, such as subfilter contribution terms or chemical source terms, is as follows:

1. Use the PIESRGAN to reconstruct ϕ_R^n from ϕ_F^n.
2. Use ϕ_R^n to estimate the unclosed terms ψ_F^n in the filtered transport equation of ϕ by evaluating the local terms with ϕ_R^n and applying a filter operator.
3. Use ψ_F^n and ϕ_F^n to advance the filtered transport equation of ϕ to ϕ_F^{n+1}.
4. Repeat steps 1–3.

2.1. Network architecture

A diagram of the PIESRGAN is depicted in Fig. 1. The generator is fed with 3-D sub-boxes of the flow fields during training and heavily uses 3-D CNN layers (Conv3D) [23] in combination with leaky rectified linear unit (LeakyReLU) layers for activation. The convolutional layers can extract increasingly complex multi-dimensional features with increasing network depth. LeakyReLU activation has the advantage of higher computational efficiency and better gradient propagation avoiding vanishing or exploding values; moreover, it introduces the leaky term on the negative input side, which prevents the so-called dead unit issue that affects standard rectified linear units (ReLUs) [24].

The residual in residual dense block (RRDB), which is introduced in ESRGAN and replaces the residual block (RB) used in previous architectures, is essential for the performance of state-of-the-art SRGANs. The RRDB contains fundamental architectural elements such as residual dense blocks (RDBs) with skip-connections, where each RDB uses dense connections inside in turn. The output from each layer within the dense block (DB) is sent to all the following layers. For PIESRGAN, RDBs are repeated multiple times using residual skip connections with the residual scaling factor β_RSF; this helps avoid instabilities in the forward and backward propagation by downscaling the residuals before additions. The motivation behind the RRDB architecture is to enable the generation of super-resolved data through a very deep network that is capable of learning and modeling all relevant complex transformations necessary to specify the required reconstruction operation.

As suggested by Wang et al. [2], all batch normalization (BN) layers of the ESRGAN architecture were deactivated for the spray application of PIESRGAN in this work. This reduces the computational cost bound to BN and was shown to improve performance compared with former single image super-resolution (SISR) models that utilized BN [25]. Furthermore, the use of BN layers can introduce distorting artifacts into generated images [2], which is absolutely undesirable in turbulence modeling, and it was found that it is not necessary for the spray application case considered in this work.

Another difference between PIESRGAN and traditional SISR applications lies in the input and output dimensions. In SISR, a generated high-resolution image contains an increased number of pixels, while the fully resolved data in turbulence contain finer structures that are enclosed in the flow. Therefore, turbulence super-resolution does not involve classical upsampling or downsampling. The input and output hold the same dimensions,

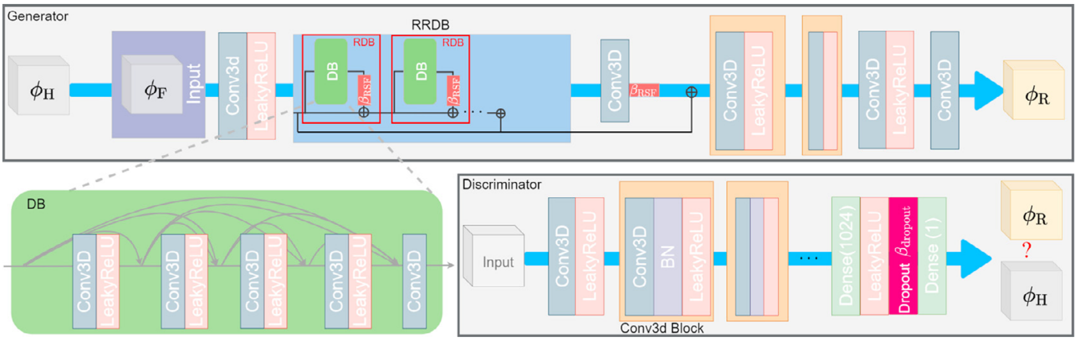

Fig. 1. Diagram of the PIESRGAN architecture.

but the output flow has more energy distributed in the high wavenumber range.

The discriminator inherits the basic CNN architecture, as also shown in Fig. 1. It consists of one Conv3d block without BN and seven Conv3d blocks with BN, followed by a fully connected layer block with dropout. LeakyReLu layers are used for activation. The blocks close to the input learn relatively simple features extracted from turbulent flows, whereas the blocks close to the output learn more complex, high-level features, like eddies/vortexes. The number of filter maps increases with depth following conventional design. The dense layer block starts with a dense layer (Dense), which projects highly dimensional output from many filter maps of the final Conv3d block onto a flat 1024 dimensional vector. The following dropout layer serves as regularization, reducing the risk of overfitting by ignoring network units during training with the probability β_{dropout}. A relativistic adversarial loss proposed by Jolicoeur-Martineau [26] is employed. Using relativistic loss as adversarial loss was shown to stabilize GAN training in different scenarios [26]. It also presumably aids learning of sharper edges and more detailed textures in SISR cases, which should also help to learn very high-frequency details in the turbulence context.

The perceptual loss proposed for the ESRGAN based on the VGG-feature space pre-trained with the ImageNet dataset is less suitable for turbulence data, as the natural image features from VGG19 may be not representative of turbulent flows. Instead, physics-informed constraints are incorporated into the loss function, guided by laws governing the physics of turbulence. More precisely, the loss function for PIESRGAN is chosen as

$$\mathcal{L} = \beta_1 L_{\text{adversarial}} + \beta_2 L_{\text{pixel}} + \beta_3 L_{\text{gradient}} + \beta_4 L_{\text{continuity}}, \tag{1}$$

where β_1, β_2, β_3, and β_4 are coefficients weighting the different loss term contributions; in this work, these coefficients were always equally scaled such that the sum of all non-zero weighting coefficients remained equal to one. Note that all loss terms are non-dimensional, since all operators and input fields used are non-dimensionalized, as explained later. $L_{\text{adversarial}}$ is the discriminator/generator relativistic adversarial loss [2], which reflects both how well the generator is able to generate high-resolution turbulence samples that look like real, DNS-obtained fully resolved turbulence flows and how well the discriminator is able to distinguish between real and generated flows. The pixel loss L_{pixel} and the gradient loss L_{gradient} are defined as the mean-squared error (MSE) of the quantity itself and of the gradient of the quantity, respectively [12]. If the MSE operator is applied on tensors, including vectors such as the velocity, it is applied to all components separately. Afterward, the resulting tensor is mapped into a scalar using the L_1-norm. $L_{\text{continuity}}$ is the physics-informed continuity loss enforcing physically plausible solutions of the reconstructed flow field in which the divergence of the velocity field should be zero for incompressible flows. If no reference solution exists, β_2 and β_3 are set to zero, reducing the loss to adversarial loss and potential continuity loss.

2.2. Implementation details

All networks were trained using cropped subboxes with size 16 × 16 × 16 from DNS and the corresponding filtered low-resolution flow fields. This box size was found to be a good compromise between memory requirement during the reconstruction step and the characteristic length scales of the flow and filter width. To map a passive scalar field combined with the velocity field, each batch with batch size 32, which is the number of samples processed before the model is updated, had the dimension 32 × 16 × 16 × 16 × 4, comprising one passive scalar channel and three velocity channels. The flow field at a given time step was divided into non-overlapping sub-boxes, which were all used for training in one epoch and accessed in random order. Based on the estimation of degrees of freedom in each sub-box compared to the number of degrees of freedom in 2-D images, 80 and 28 layers

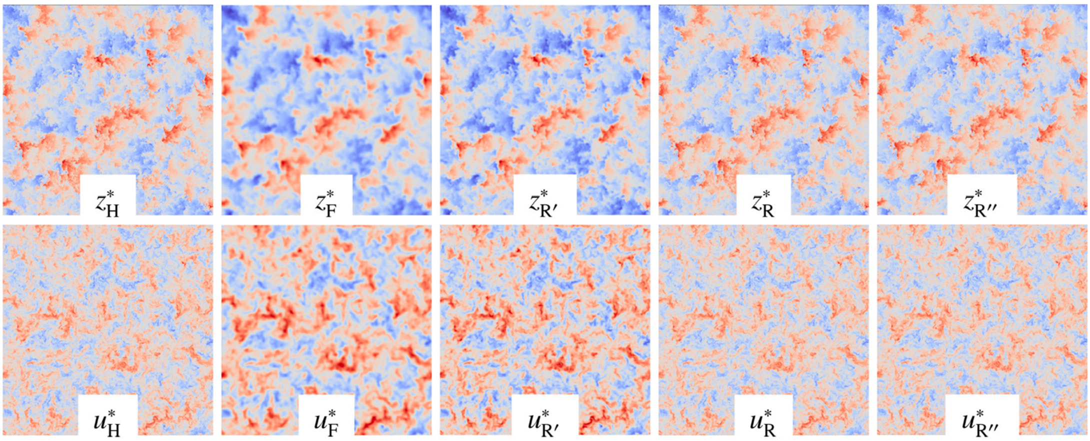

Fig. 2. Visualization of 2-D slices of the dimensionless passive scalar z^* and the dimensionless velocity component u^* for the time step with Taylor microscale-based Reynolds number Re_τ of about 88.

were chosen for the generator and discriminator, respectively. RMSProp, which relies on the stochastic gradient descent (SGD) approach, was used as the optimizer. Before using them for training and reconstruction, all fields were zero mean-centered and rescaled with their root-mean-square deviation (RMSD) value, resulting in dimensionless quantities (denoted by an asterisk). For homogeneous isotropic vector fields, a single averaged RMSD value computed as the root of the average of the squared RMSD values of all components was used for all components. Temporal and spatial operators were non-dimensionalized with the RMSD value of the initial velocity field in the DNS $u^{\mathrm{init}}_{\mathrm{RMSD}}$ and the wavenumber magnitude κ_{p}, which corresponds to the peak in the initial energy spectrum in the DNS. Without centering and rescaling all fields, the loss function usually diverged during training.

To increase the reproducibility of this work and clarify more technical details, the implemented PIESRGAN was uploaded to GIT (https://git.rwth-aachen.de/Mathis.Bode/PIESRGAN.git).

2.3. Training strategy

Many industrially relevant applications are operated at very high Reynolds numbers that are not accessible by DNS. Thus, training the network only with the DNS data of the relevant Reynolds number range is not possible. This raises the question whether a network trained with DNS data of lower Reynolds numbers is general enough to give also good results at higher Reynolds numbers, i. e., whether it has an extrapolation capability.

In the a priori test (cf. Subsection 2.4), it was seen that training the network only with DNS data leads to a lack of accuracy for Reynolds numbers falling outside the training range. Therefore, in this work, the training was extended by introducing a second step. After training the generator and discriminator simultaneously with DNS data (“H”) and corresponding filtered data (“F”) only, the generator was further trained and updated using filtered data (“$\tilde{\mathrm{F}}$”), which were generated for a wider range of Reynolds numbers with LES without subfilter closing, which can be computed at low computational cost. Corresponding “H” data did not exist, and the discriminator was not further updated. Note that, as mentioned before, the loss function reduces for this second learning phase, as the evaluation of loss terms related to DNS data is no longer possible. Thus, the loss is driven mainly by the part of the adversarial loss that corresponded to correctly recognizing generated flow samples and by the physics-informed continuity constraint.

Another critical challenge during the training is the search for optimal hyperparameters. The training was started with the hyperparameters given by Wang et al. [2]. However, the weighting coefficients in the loss function were initially chosen such that all loss terms had a similar order of magnitude after a few numbers of epochs. Afterward, the weighting coefficient β_4 was increased until the error in the non-dimensionalized continuity equation based on the zero mean-centered and rescaled velocity field was comparable to the order of magnitude of the error introduced by the numerical approximation of the divergence operator. Subsequently, the network parameters were tuned by estimating the sensitivities around the starting values and choosing locally optimal values. Finally, the remaining weights of the loss function were adjusted. Since it was not sufficient to judge the quality of hyperparameter combinations only based on the loss function, other measurements, such as the spectrum,

Table 1
Overview of the PIESRGAN hyperparameters. The given ranges represent the sensitivity intervals with acceptable network results. The central values were finally used in this work.

β_1	$[0.2 \times 10^{-5}, 0.6 \times 10^{-4}, 0.8 \times 10^{-4}]$
β_2	$[0.79327, 0.88994, 0.91812]$
β_3	$[0.04, 0.06, 0.15]$
β_4	$[0.01, 0.05, 0.06]$
β_{RSF}	$[0.1, 0.2, 0.3]$
β_{dropout}	$[0.2, 0.4, 0.5]$
$l_{\mathrm{generator}}$	$[1.2 \times 10^{-6}, 4.5 \times 10^{-6}, 5.0 \times 10^{-6}]$
$l_{\mathrm{discriminator}}$	$[4.4 \times 10^{-6}, 4.5 \times 10^{-6}, 8.5 \times 10^{-6}]$

were also considered. Table 1 gives a summary of hyperparameters used for the application case in this work, including the learning rates for the generator training and discriminator training, $l_{\mathrm{generator}}$ and $l_{\mathrm{discriminator}}$.

2.4. A priori testing

In this work, one of the largest existing decaying turbulence DNS datasets [27] was used for training and testing the PIESRGAN. The dataset features periodic boxes of homogeneous isotropic turbulence with Reynolds numbers based on the Taylor microscale Re_τ of up to 88, simulated on 4096^3 mesh points. The first data time step investigated was defined to lie in the self-similar range of the flow, as indicated with $t^{*\mathrm{start}}$ in Fig. 4. Before the training, the data was filtered to obtain combinations of "H" and "F". The PIESRGAN could reconstruct the data within the trained Reynolds number range well. To test the extrapolation capability of the network, the first time step of the DNS data was used only for testing, not for training. As the Reynolds number reduces over time for the decaying turbulence case, skipping the first time step of the data resulted in a highest Reynolds number of about 75 for the training, while testing was performed with a Reynolds number of 88. The results are shown in Fig. 2 as "R′" for the dimensionless passive scalar z^* and one dimensionless velocity component u^*. It is obvious that the network adds insufficient small-scale structures to the flow - maybe because it had never seen such a high Reynolds number before, i. e., earlier, it never needed to add such small structures to the flow. The column labeled "R" shows the results of a network additionally trained with "$\tilde{\mathrm{F}}$" data, featuring Re_τ of up to 250. The reconstruction results are much better, and the visual agreement with the DNS data is almost perfect. One reason for this could be that the "$\tilde{\mathrm{F}}$" data merely modified all weights in the network, which results in higher subfilter contributions for all Reynolds numbers, randomly leading to good reproduction for the target Reynolds number but worse results for the others. This would contradict the idea that the neural network used the new data to "really learn" the results of the target Reynolds number by means of the adversarial loss. Therefore, the PIESRGAN was alternatively trained with the "H"/"F" dataset and additionally a dataset "$\tilde{\mathrm{F}}'$", featuring only Re_τ of up to 200. These results are shown in Fig. 2 with "R″", and the agreement is as good as before, which indicates that the network really learned to reproduce the higher Reynolds number data. This was also confirmed by analyzing the results for "R′" and "R" within the Reynolds number training range, which did not differ (these results are not shown here). The same result was also observed for the other two velocity components, which are not visualized in Fig. 2. Note that higher Reynolds number data can be chosen arbitrarily as long as the maximum Re_τ is higher than the target Re_τ, which was 88 for the a priori test. However, the large Re_τ of 250 was selected to enable usage of the trained network for the ECN Spray A case (cf. Section 3).

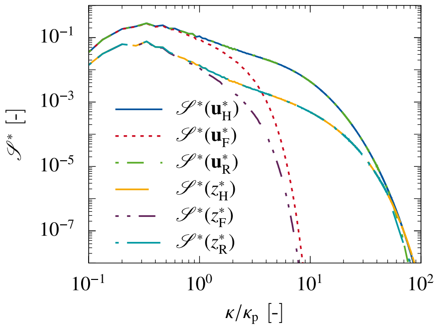

Fig. 3. Dimensionless spectra evaluated on DNS data, filtered data, and reconstructed data for the dimensionless velocity vector $\mathbf{u}^*$ and passive scalar z^* for the time step with Reynolds number of about 88.

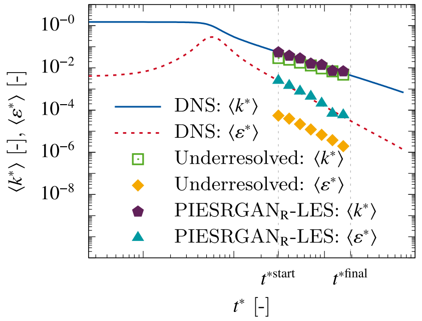

Fig. 4. Temporal evolution of the ensemble-averaged dimensionless turbulent kinetic energy $\langle k^* \rangle$ and ensemble-averaged dimensionless dissipation rate $\langle \varepsilon^* \rangle$.

In addition to the visual evaluation, Fig. 3 presents the non-dimensionalized spectra (denoted by $\mathscr{S}^*$) computed with "H", "F", and "R" data. It shows that also the statistical agreement between DNS and reconstructed data is very good. Only for very high wavenumber magnitudes κ, the reconstructed flow field slightly differs from the DNS data. Note that the spectra based on the velocity use all three velocity components. Therefore, $\mathscr{S}^*(\mathbf{u}^*)$ with bold notation for vectors is shown. Moreover, the average maximum error evaluated over 1000 reconstructed realizations in the non-dimensionalized continuity equation was 4.33×10^{-8} with a standard deviation of 2.17×10^{-8}. This emphasizes the strong impact of the physics-informed continuity loss term, as the error in the non-dimensionalized continuity equation was on average in the order of 1×10^{-2} for networks trained with continuity loss term disabled (i. e., $\beta_4 = 0$).

The results presented in Figs. 2 and 3 indicate that the PIESRGAN is able to learn universal key features of turbulence with the adversarial loss, enabling the correct prediction of statistics of higher Reynolds number flows, only seeing filtered data. This is a significant advantage to networks that have to rely on supervised learning with labeled data. How the network is able to detect the target Reynolds number from the provided fields with zero mean is an open question and should be addressed in more detail in future work. The DL may exploit the scale similarity of small-scale turbulence, which is a universal feature at sufficiently high Reynolds numbers [28], to predict flow fields at different Reynolds numbers. For a posteriori testing and application, the trained network corresponding to the results labeled with “R” in Figs. 2 and 3 is denoted by PIESRGAN$_\mathrm{R}$.

2.5. A posteriori testing

Before using the trained network PIESRGAN$_\mathrm{R}$ in a complex reactive turbulent flow, an a posteriori test was performed with respect to the decaying turbulence data. For the test, filtered data of the early time step $t^{*\mathrm{start}}$ of a realization of the decaying turbulence DNS data, which was not used during the training, were used as the initial flow field and advanced over time t according to the steps outlined in the beginning of this section. To keep the filter width of the data consistent with the training data, the DNS data of size 4096^3 were filtered to a 64^3 mesh. The time step size of the LES was increased compared to the DNS, and approximated ensemble averages were computed over all grid points of a single time step. Fig. 4 compares the decay of the ensemble-averaged dimensionless turbulent kinetic energy $\langle k^* \rangle$ and the ensemble-averaged dimensionless dissipation rate $\langle \varepsilon^* \rangle$ evaluated during the DNS and the a posteriori test with PIESRGAN$_\mathrm{R}$ as LES model. All quantities were non-dimensionalized using $u_\mathrm{RMSD}^\mathrm{init}$ and κ_p. To assess the effect of the subfilter model, the results for an underresolved simulation computed on the LES mesh without any subgrid model are also plotted. The good agreement between DNS and PIESRGAN$_\mathrm{R}$-LES is remarkable, while the underresolved simulation underpredicts both quantities and the absolute value of the decay rate of $\langle k^* \rangle$. During the decay, the Kolmogorov length scale and the integral length scale increases with time, following a power law. This implies that the number of wavenumbers needing closure decreases during the decay. The PIESRGAN accounts for this change of the relative relevance of the subfilter closure, which underlines its ability to model small-scale turbulence.

3. Application

One prominent example of turbulent reactive flow is the Spray A case defined by the ECN [22], with Re$_\tau$ of up to 235, which is chosen here to demonstrate the usage of the trained PIESRGAN$_\mathrm{R}$ for combustion. PIESRGAN$_\mathrm{R}$ was used as LES-subfilter model for the subfilter turbulent flux in the equation of the mixture fraction Z and for the subfilter Reynolds stresses in the momentum equations. More precisely, the same reactive and inert conditions and simulation setup as discussed in Davidovic et al. [29] were computed, using the chemical mechanism of Yao et al. [30] and a multiple representative interactive flamelets (MRIF) model. Details of the simulation setup and numerics can be found in former publications [29,31–35]. Compared to the simulations performed by Davidovic et al. [29], a coarser mesh was used in this work to emphasize the effects of the subfilter model, resulting in a minimum grid spacing of $100\,\mu$m close to the nozzle. A visualization of a reactive LES with PIESRGAN$_\mathrm{R}$ as subfilter model for mixture fraction and velocity can be found in the supplementary material video S1. Note that the PIESRGAN$_\mathrm{R}$ was also used to evaluate the mixture fraction variance of the reconstructed mixture fraction field. It could have also been used to compute the subfilter probability density function (PDF) of Z used as part of the MRIF model, but instead a classical presumed shape beta-PDF was used here. Furthermore, the ignition delay time, defined as the time when the Favre-filtered OH mass fraction $\widetilde{Y}_\mathrm{OH}$ reaches 2% of its maximum value $\widetilde{Y}_\mathrm{OH}^\mathrm{max}$ (all-time) for the first time, was evaluated as the average value with a standard deviation of 0.385 ± 0.007 ms (based on five realizations), which reasonably agreed with the experimental values of about 0.4 ms [22]. Averaged over the five realizations, the time-averaged lift-off length (LOL) of the flame during the statistically stationary phase defined with the 14 %-$\widetilde{Y}_\mathrm{OH}^\mathrm{max}$ (instantaneous)-contour was found to be 15.6 ± 0.1 mm. This is also close to experimental

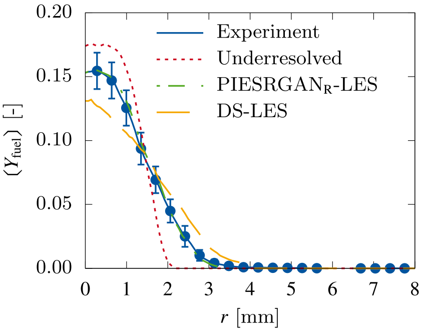

Fig. 5. Temporally and circumferentially averaged fuel mass fraction $\langle Y_{\mathrm{fuel}} \rangle$ evaluated 18.75 mm downstream from the nozzle and plotted against the radial distance from the spray axis r.

data for this nozzle, which report an LOL of about 16.1 mm [22].

Non-premixed combustion, including emission formation, is heavily affected by the mixing of fuel and oxidizer stream. In order to assess the effect of the subfilter modeling on mixing, the fuel mass fraction Y_{fuel} is evaluated 18.75 mm downstream from the nozzle for the inert conditions. It is temporally and circumferentially averaged during the statistically stationary phase labeled as $\langle Y_{\mathrm{fuel}} \rangle$ and plotted against the radial distance from the spray axis r in Fig. 5 for the PIESRGAN$_{\mathrm{R}}$-LES, an LES with dynamic Smagorinsky model [29] (denoted by 'DS-LES'), an underresolved simulation without any subgrid model, and experimental data [36]. The mixing of the PIESRGAN$_{\mathrm{R}}$-LES is weaker than that of the DS-LES and in better agreement with the experimental data. This indicates that the PIESRGAN$_{\mathrm{R}}$ is a robust and accurate model, which is impressive considering that it was trained only with homogeneous isotropic turbulence data. The underresolved simulation underpredicts the mixing due to the lack of turbulent transport and is not robust, as fluctuations accumulate on the grid-scale, blowing up the simulation. This shows the importance of subfilter models for this application case. Note that the PIESRGAN$_{\mathrm{R}}$-LESs were run without any clipping, which weakens the hypothesis that data-driven models are dangerous to use in real simulations as extreme predictions might crash the simulation. Computationally, the PIESRGAN$_{\mathrm{R}}$-LES was more expensive than the DS-LES. However, with the rapid improvements in the field of DL on GPUs, this could change in the near future.

4. Conclusions

This work presents a novel GAN-based subfilter modeling approach, which employs unsupervised DL with a combination of super-resolution adversarial and physics-informed losses to accurately predict subfilter statistics in a wide Reynolds number range. The PIESRGAN was trained with some of the largest existing decaying turbulence data. It was found that successive training with fully resolved and underresolved data increases the generalization capability of the network. Moreover, it was shown that the trained network gives good results in a priori and a posteriori tests, and the importance of the physics-informed loss term based on the continuity equation was demonstrated. The advantages of PIESRGAN-LES over LES with dynamic Smagorinsky model were discussed by applying it to the ECN Spray A case with inert and reactive conditions. Although some aspects of the network are not fully understood yet, and the data processing speed needs to be improved, this work emphasizes the enormous potential of data-driven models for reactive flows.

In this work, the GAN-method was applied to model subfilter terms for momentum and scalar mixing. The application to reactive scalar fields to close the chemical source term might also be beneficial, but it is challenging for different reasons. For instance, for fast chemistry, the source term depends on the "very smallest" scales, which means that these need to be correctly predicted for multiscalar fields. It will be interesting to assess the potential in future work.

Declaration of Competing Interest

None.

Acknowledgments

The authors acknowledge the funding received from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement No 695747. Moreover, the computing time grants for the project JHPC55 by the JARA-HPC Vergabegremium provided on the JARA-HPC Partition part of the supercomputer JURECA at Jülich Supercomputing Center, Forschungszentrum Jülich, and on the national supercomputer Cray XC40 at the High-Performance Computing Center Stuttgart (HLRS) under the grant number GCS-MRES are acknowledged.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.proci.2020.06.022

References

[1] C. Dong, C.C. Loy, K. He, X. Tang, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 184–199.

[2] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, C. Loy, *LNCS* 11133 (2019) 63–79.

[3] H. Greenspan, B. Van Ginneken, R.M. Summers, *IEEE Trans. Med. Imaging* 35 (5) (2016) 1153–1159.

[4] G. Hinton, L. Deng, D. Yu, et al., *IEEE Signal Process. Mag.* 29 (2012).

[5] O. Vinyals, I. Babuschkin, W.M. Czarnecki, et al., *Nature* 575 (2019) 350–354.

[6] E.J. Parish, K. Duraisamy, *J. Comput. Phys.* 305 (2016) 758–774.

[7] P. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, R. Vinuesa, *Phys. Rev. Fluids* 4 (5) (2019) 054603.

[8] R. Maulik, O. San, *J. Fluid Mech.* 831 (2017) 151–181.

[9] M. Bode, M. Gauding, J. Göbbert, B. Liao, J. Jitsev, H. Pitsch, *LNCS* 11203 (2018) 614–623.

[10] J. Kutz, *J. Fluid Mech.* 814 (2017) 1–4.

[11] C.J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, T. Poinsot, *Combust. Flame* 203 (2019) 255–264.

[12] M. Bode, M. Gauding, K. Kleinheinz, H. Pitsch, *LNCS* 11887 (2019) 541–560.

[13] M. Bode, M. Gauding, J. Jitsev, H. Pitsch, Development of physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling, arXiv preprint (2021).

[14] M. Raissi, P. Perdikaris, G. Karniadakis, *J. Comput. Phys.* 378 (2019) 686–707.

[15] I. Banerjee, M.G. Ierapetritou, *Combust. Flame* 144 (3) (2006) 619–633.

[16] M. Ihme, C. Schmitt, H. Pitsch, *Proc. Combust. Inst.* 32 (1) (2009) 1527–1535.

[17] M. Bode, N. Collier, F. Bisetti, H. Pitsch, *Combust. Theor. Model.* 23 (4) (2019) 674–699.

[18] A. Seltz, P. Domingo, L. Vervisch, Z.M. Nikolaou, *Combust. Flame* 210 (12) (2019) 71–82.

[19] Q. Wang, M. Ihme, *Combust. Flame* 204 (6) (2019) 341–355.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.

[21] C. Ledig, L. Theis, F. Huszár, et al., in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4681–4690.

[22] Engine combustion network: https://ecn.sandia.gov, 2019.

[23] A. Krizhevsky, I. Sutskever, G.E. Hinton, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[24] A.L. Maas, A.Y. Hannun, A.Y. Ng, in: Proceedings of the 30th International Conference on Machine Learning, 30, 2013.

[25] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 136–144.

[26] A. Jolicoeur-Martineau, arXiv: 1807.00734 (2018).

[27] M. Gauding, L. Wang, J.H. Goebbert, M. Bode, L. Danaila, E. Varea, *Comput. Fluids* 180 (2019) 206–217.

[28] L. Djenidi, R.A. Antonia, S.L. Tang, *J. Fluid Mech.* 864 (2019) 244–272.

[29] M. Davidovic, T. Falkenstein, M. Bode, L. Cai, S. Kang, J. Hinrichs, H. Pitsch, *Oil Gas Sci. Technol.* 72 (29) (2017).

[30] T. Yao, Y. Pei, B.-J. Zhong, S. Som, T. Lu, K. Hong Luo, *Fuel* 191 (2017) 339–349.

[31] M. Bode, F. Diewald, D. Broll, J. Heyse, et al., *SAE Technical Paper 2014-01-1427*, 2014, https://doi.org/10.4271/2014-01-1427.

[32] M. Bode, T. Falkenstein, V. Le Chenadee, S. Kang et al., *SAE Technical Paper 2015-01-0949*, 2015, https://doi.org/10.4271/2015-01-0949.

[33] M. Bode, T. Falkenstein, M. Davidovic, et al., *SAE Int. J. Fuels Lubr.* 10 (2) (2017a) 380–393.

[34] M. Bode, M. Davidovic, H. Pitsch, in: High-Performance Scientific Computing, Springer Int. Publishing, 2017b, pp. 96–108.

[35] M. Bode, M. Davidovic, H. Pitsch, in: High-Performance Computing in Science, Springer Nature, 2019, pp. 185–207.

[36] L.M. Pickett, C.L. Genzale, J. Manin, *Atom.Sprays* 25 (2015) 425–452.