Influence of adversarial training on super-resolution turbulence reconstruction

Ludovico Nista ⬡* and Heinz Pitsch ⬡

Institute for Combustion Technology, RWTH Aachen University, Aachen, 52056, Germany

Christoph D. K. Schumann ⬡

Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, United Kingdom

Mathis Bode ⬡

Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Jülich, 52425, Germany

Temistocle Grenga ⬡

Department of Aeronautics and Astronautics, Faculty of Engineering and Physical Sciences,
University of Southampton, Southampton, SO17 1BJ, United Kingdom

Jonathan F. MacArt ⬡

Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, Indiana 46556, USA

Antonio Attili ⬡

School of Engineering, Institute for Multiscale Thermofluids,
University of Edinburgh, Edinburgh, EH9 3FD, United Kingdom

Supervised super-resolution deep convolutional neural networks (CNNs) have gained significant attention for their potential in reconstructing velocity and scalar fields in turbulent flows. Despite their popularity, CNNs currently lack the ability to accurately produce high-frequency and small-scale features, and tests of their generalizability to out-of-sample flows are not widespread. Generative adversarial networks (GANs), which consist of two distinct neural networks (NNs), a generator and discriminator, are a promising alternative, allowing for both semisupervised and unsupervised training. The difference in the flow fields produced by these two NN architectures has not been thoroughly investigated, and a comprehensive understanding of the discriminator's role has yet to be developed. This study assesses the effectiveness of the unsupervised adversarial training in GANs for turbulence reconstruction in forced homogeneous isotropic turbulence. GAN-based architectures are found to outperform supervised CNNs for turbulent flow reconstruction for in-sample cases. The reconstruction accuracy of both architectures diminishes for out-of-sample cases, though the GAN's discriminator network significantly improves the generator's out-of-sample robustness using either an additional unsupervised training step with large eddy simulation input fields or a dynamic selection of the most suitable up-sampling factor. These enhance the generator's ability to reconstruct small-scale gradients, turbulence intermittency, and velocity-gradient probability density functions. Conversely, the supervised super-resolution CNN network lacks the capability to reconstruct these statistics. The extrapolation capability of the GAN-based model is demonstrated for out-of-sample flows at higher Reynolds numbers. Based on these findings, incorporating

---

*l.nista@itv.rwth-aachen.de

064601-1

discriminator-based training is recommended to enhance the reconstruction capability of super-resolution CNNs.

I. INTRODUCTION

Recent advances in experimental diagnostics and direct numerical simulation (DNS) capabilities have enabled the generation of increasingly high-fidelity turbulent flow fields. Doing so requires expensive assessments and significant computational resources due to the chaotic, multiscale nature of turbulence, making them impractical for engineering applications. Consequently, high-fidelity reconstruction of turbulent flows from limited data has been a longstanding concern. In this context, the ability to restore high-resolution (HR) from low-resolution (LR) fields is an attractive proposition. For example, accurately representing small-scale turbulent features is necessary to capture mixing at these scales, which could enable more accurate subfilter-scale (SFS) closure models for large eddy simulations (LES) [1]. Similarly, in particle image velocimetry (PIV), near-wall resolution limitations constrain the ability to capture intricate flow structures close to boundaries, where high-resolution information is crucial for understanding boundary layer dynamics [2].

Significant strides in modern deep-learning frameworks have motivated the development of data-driven super-resolution (SR) methods for turbulent flows, which are promising compared to traditional SR techniques [3,4]. Their objective is to design suitable neural network architectures to upsample LR (ϕ_{LR}) fields to HR fields (ϕ_{HR}). The resulting model should accurately invert an unknown filter operator $\mathcal{G}$,

$$\phi_{\mathrm{HR}} \approx \phi_{\mathrm{SR}} = \mathcal{G}_l^{-1} * \phi_{\mathrm{LR}} = \mathcal{G}_l^{-1} * \mathcal{G} * \phi_{\mathrm{HR}}, \tag{1}$$

where $\mathcal{G}_l^{-1}$ is an lth-order approximate inverse of $\mathcal{G}$. The quality of the approximate inverse operator depends heavily on the choice of network architecture and training methodology and is not universal for implicit kernel functions.

Deep convolutional neural networks (CNNs) have been applied for SR, with Fukami *et al.* [5–7] pioneering their use for chaotic flows. They used a conventional super-resolution CNN to reconstruct turbulent velocity and vorticity fields from LR input data obtained from experimental data and numerical calculations. Using a similar architecture with residual layers [8], Pant and Farimani [9] trained a deep CNN on forced homogeneous isotropic turbulence (HIT) to super-resolve filtered DNS fields and addressed the tradeoff between resolution (fidelity) and computational complexity. Liu *et al.* [10] used a deep CNN model for SR reconstruction of 3D HIT, finding that training on time-series velocity fields improved the model's SR reconstruction capabilities. Recently, Zhou *et al.* [11] developed a turbulence volumetric super-resolution (TVSR) model based on CNNs, trained on various Reynolds numbers, that was robust and accurate for different Reynolds numbers, though only when coupled with an approximate deconvolution method (ADM).

While SR architectures have been shown to be more accurate than analytical models for in-sample predictions (i.e., tested on data that statistically matches training data) [11,12], the current understanding of SR mainly involves results obtained from supervised training of CNN models, which require labeled LR and HR data for training. The necessarily large amounts of labeled data for this, often from DNS, can be impractical to obtain. Supervised SR-CNN-based architectures do not typically generalize to out-of-sample flow conditions or domains, for they are optimized for specific data sets and objective functions, hence they remain largely untested for out-of-sample inputs. Furthermore, supervised CNN architectures do not accurately predict high-resolution details, resulting in a loss of high-wave number features and producing blurry super-resolved fields. This is a significant limitation for small-scale reconstruction requiring accurate recovery of high-frequency detail.

To address the drawbacks of the supervised CNNs training strategy for SR, generative adversarial networks (GANs) have been employed [13] that, unlike CNNs, do not rely solely on user-defined objective functions for training. Instead, GANs' super-resolving generator networks, based on CNNs, are coupled with discriminator networks that compete in semisupervised learning. In GAN-based SR, the generator produces higher-resolution versions of low-resolution fields (e.g., filtered DNS or LES), while the discriminator network attempts to differentiate between the generated high-resolution field and the real, "ground truth" high-resolution field (e.g., the DNS itself). During training, the generator learns to produce samples that are indistinguishable from genuine high-resolution data. The discriminator in turn learns to judge the authenticity of the samples. As a result, both networks improve each other during the adversarial training process.

Deng *et al.* [14] compared super-resolution GAN (SRGAN) [15] and enhanced SRGAN (ESRGAN) [16] approaches, initially developed for image reconstruction, for the reconstruction of the flow around tandem cylinders by upscaling two-dimensional turbulence. The ESRGAN architecture outperformed the SRGAN in mean-flow metrics and fluctuation distributions. Through the use of GAN architectures with and without physical loss functions (based on mass and momentum conservation), Lee and You [17] studied the prediction of flow around a cylinder in laminar conditions. One should note that this approach barely differs from DNS and is thus, by default, computationally prohibitive. In reconstructing temporal data from a large time-step interval, only the physics-based loss function adequately constrained the solution space for accurate prediction of the resolved flow motion. Xu *et al.* [18] presented an innovative architecture that leverages transformers in conjunction with a GAN; they found this architecture to effectively reconstruct high-resolution turbulence fields for isotropic and anisotropic flows. Kim *et al.* [4] addressed the challenge of reconstructing small-scale turbulence for sparsely paired LR and HR data, adopting a cycle-consistent generative adversarial network (CycleGAN). The architecture successfully reconstructs HR flow fields with DNS-quality statistics from the LES data. To drive the (unsupervised) learning process, they designed a loss function based on the assumption that the filtered-DNS (F-DNS) fields share a similar distribution with LES data. Consequently, their approach did not use unsupervised adversarial training.

Bode *et al.* [19,20] introduced a physics-informed ESRGAN (PIESRGAN) for SFS turbulence reconstruction incorporating a loss function based on the continuity-equation residual alone. Although their model was only trained on HIT data, it was able to improve scalar-mixing predictions in a reacting jet. Nista *et al.* [21,22] then investigated the generalization capability of a similar GAN architecture by evaluating the model's extrapolability to higher and lower Reynolds numbers than those used for training. They found that the ratio between the LES filter width and the Kolmogorov scale must be preserved for adequate generalization (i.e., a fixed SR upsampling window). Grenga *et al.* [23] investigated the ability of similar GAN models to recognize and reconstruct gradient and countergradient transport in low- and high-Karlovitz-number combustion regimes [24,25]. In this case, only a GAN trained for both data sets adequately reconstructed the subfilter scales for both combustion regimes.

While numerous studies have recognized the superior performance of GANs for super-resolution reconstruction over conventional supervised CNNs [4,17,26–28], the underlying reasons for this advantage have not been thoroughly explored. Possibilities include the flexibility in the training strategy (supervised and partially unsupervised), the use of adversarial (unsupervised) training leveraging unpaired training data, and the exploitation of generative models to model distributions, thereby improving generalization to out-of-sample data. GANs are not without drawbacks and may exhibit training instability and mode collapse, which could hinder convergence of the generator [29]. Given these and the computational cost of training GANs, a critical evaluation of the need for SR-GANs and the influence of the discriminator on generalizability to both in- and out-of-sample flows with respect to different kernels and sizes and different flow configurations is necessary. Moreover, the performance of GAN-based models to reconstruct spatial features and correlations, including gradients, intermittency, and probability density function tails, has not been thoroughly examined.

TABLE I. Simulation parameters for training and testing data sets. N is the number of mesh points, Re_t is the turbulent Reynolds number, L_x/L_{11} is the number of integral scales within the computational domain, dx/η is the mesh resolution relative to the Kolmogorov microscale η, and k_{max} is the largest wave number represented. "Train" and "test" flags indicate which data were used in model training and testing, respectively.

Case	N	Re_λ	Re_t	L_x/L_{11}	dx/η	k_{max}	Train	Test
Re90	256^3	90	920	5.26	1.98	128	●	●
Re130	512^3	130	3320	5.26	1.86	256		●
Re350	1024^3	350	16182	5.26	1.97	512		●

This study investigates and quantifies the role of the GAN discriminator in reconstructing turbulent fields while considering the benefits and limitations of the GAN approach. To evaluate the effectiveness of adversarial training, a supervised deep-CNN model (TSResNet; Sec. III A) is directly compared to a semisupervised/unsupervised GAN (PIESRGAN; Sec. III B). The same CNN architecture, physics-inspired loss functions, and training data are used for the TSResNet and PIESRGAN models. Hence, the sole difference between the two lies in the training strategy. In particular, the ability to accurately reconstruct small-scale structures of turbulence, consistent with the fundamental theory of turbulence, is investigated. Furthermore, while most studies have focused on in-sample reconstruction within a narrow range of flow conditions, we explore both models' generalizability to out-of-sample filtered fields for different filter sizes, filter kernels, and Reynolds numbers. Our aim is to develop an understanding of the consistency of the reconstructed fields with the original flow while avoiding the generation of spurious and/or unphysical fields and preserving the fundamental turbulence characteristics. Particular emphasis is placed on the improvements of these metrics brought about by adversarial training.

II. DATA SETS AND PREPROCESSING

Three forced HIT DNS data sets at Taylor-microscale Reynolds numbers $Re_\lambda = 90$, $Re_\lambda = 130$, and $Re_\lambda = 350$ are considered. These are computed using the CIAO code, which uses second-order central differences, staggered meshes, and second-order implicit time advancement to solve the incompressible Navier-Stokes equations [30]. The calculations have mesh sizes $256 \times 256 \times 256$ points for the low Re, $512 \times 512 \times 512$ points for intermediate Re, and $1024 \times 1024 \times 1024$ points for the high Re. The linear forcing $\mathbf{f} = A\,\mathbf{u}$ proposed by Lundgren [31] is applied, where A is a forcing parameter inversely proportional to the eddy turnover time. Table I reports the simulation parameters for the three data sets. SR training used only the lowest Re DNS (*Re90*), with the higher Re DNSs (*Re130* and *Re350*) being reserved for testing.

After the lowest Re simulation reached a statistically stationary state, 160 snapshots of the 3D velocity vector $[u_i = (U, V, W)^T]$ were extracted every five eddy-turnover times for training. To obtain low-resolution data, all available snapshots of the instantaneous velocity were filtered using explicit filter kernels of width $\Delta = 4\,dx$, corresponding to $8\,\eta$. Our analysis is limited by memory requirements associated with the upsampling factor. As demonstrated later, the filter size considered places the cutoff wave number into the inertial range. As the primary goal is to demonstrate the effectiveness of adversarial training for SR reconstructions by comparing different training strategies, the advantage of adversarial contribution is apparent even at these upsampling factors. The training, validation, and testing data sets were composed of high-resolution (DNS) data, and the corresponding low-resolution (filtered-DNS) data. The spatial box, Gaussian, and spectrally sharp filter kernels are defined as

$$\mathcal{G}_{box}(\boldsymbol{x}) = \begin{cases} \frac{1}{\Delta} & \text{if}|\boldsymbol{x}| \leqslant \frac{\Delta}{2} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

$$\mathcal{G}_{\text{Gaussian}}(\boldsymbol{x}) = \left(\frac{6}{\pi \Delta^2}\right)^{\frac{1}{2}} \exp\left(-\frac{6(\boldsymbol{x})^2}{\Delta^2}\right), \tag{3}$$

$$\mathcal{G}_{\text{spectral}}(\boldsymbol{x}) = \frac{\sin\left[\pi(\boldsymbol{x})/\Delta\right]}{\pi(\boldsymbol{x})/\Delta}, \tag{4}$$

and the F-DNS fields are obtained by a discrete downsampling operation, applied independently of the filter kernel, defined as

$$\mathbf{f}_{\text{downsampling}} : \mathbb{R}^{\Omega} \to \mathbb{R}^{\Omega/\Delta}, \quad \bar{\mathbf{u}}_{\text{F-DNS}} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \mathcal{G} * \mathbf{u}(\Delta i, \Delta j, \Delta k), \tag{5}$$

where N is the number of mesh points of the high-resolution field and Ω is the corresponding three-dimensional field. The latter operation ensures that the F-DNS fields have the same dimensionality as the corresponding LES fields. A comparison of filter kernels (box, Gaussian, and spectrally sharp) is given in Sec. V A.

Loading the entire training data set is impractical given memory limitations. To alleviate this, we extract subdomains randomly from each snapshot of the full domain. The first 120 snapshots were used for training, while the last 40 frames, equally divided, were used for validation and testing (training/validation ratio = 0.75/0.25). The ongoing validation was then performed on 20 in-sample fields not used for training. The last 20 snapshots were employed to obtain averaged results. The time separation between training/validation samples and testing samples spans more than 10 large-eddy turnover times, thus the two sets of samples should be statistically uncorrelated. Both networks, TSResNet and PIESRGAN (introduced in Sec. III), were trained using subboxes of the original computational domain with a size of $128\,\eta \times 128\,\eta \times 128\,\eta$, corresponding to subboxes of $64 \times 64 \times 64$ and $16 \times 16 \times 16$ for high- and low-resolution fields, respectively. The size of these subboxes was found to be a good compromise between memory requirements and the inherent length scale of the turbulent flow. It is important to emphasize that the dimensions of the subboxes were constrained by both η and the integral length scale l_t. Specifically, it is ensured that at least one l_t is contained in these subboxes. Using those settings, 6400 subboxes were employed for the training. During the evaluation of testing samples, the entire test sample is reconstructed continuously, without employing cropped reconstruction boxes as is done for training. The velocity components of high-resolution data and low-resolution data used for training and testing were normalized by the global maximum and minimum of the DNS [i.e., $(U_{\text{F-DNS}}, V_{\text{F-DNS}}, W_{\text{F-DNS}}) \in [0, 1]$ and $(U_{\text{DNS}}, V_{\text{DNS}}, W_{\text{DNS}}) \in [0, 1]$] to improve the network's performance [32].

The large data set and deep convolutional frameworks (implemented in TensorFlow [33]), required training parallelization; this was done using the Horovod library [34]. Calculations were performed on the Jülich DEEP-EST cluster (DEEP-DAM partition) using four nodes, each with one NVIDIA V100 32 GB GPU. This gave an overall speed-up close to a factor of four versus single-GPU training. A minibatch of eight subboxes per GPU and the ADAM optimizer, with an initial learning rate of 10^{-4}, were chosen due to memory constraints and previous evidence of the success of this combination [9,17,35]. "Mixed-precision" training is common in machine learning for its low computational cost [36] but can be detrimental to predictive accuracy in scientific computing. We use single-precision arithmetic for both weight definition and loss computations, following the results of Hrycej *et al.* [37]. Details are provided in the Appendix. Moreover, during the evaluation of the testing samples, CPUs with access to a significantly larger memory pool are employed to overcome memory limitations.

For unsupervised training, LES calculations of forced HIT are conducted using the same DNS solver configuration albeit for coarser grid resolutions (equivalent to the discrete downsampling factor applied to the DNS mesh). The unresolved scales are closed using the dynamic Smagorinsky model [38,39] [Eq. (11)], or their influence is not explicitly modeled ("implicitly modeled" LES). To extrapolate a given model to higher Re (utilizing the *Re130* and *Re350* data sets for out-of-sample
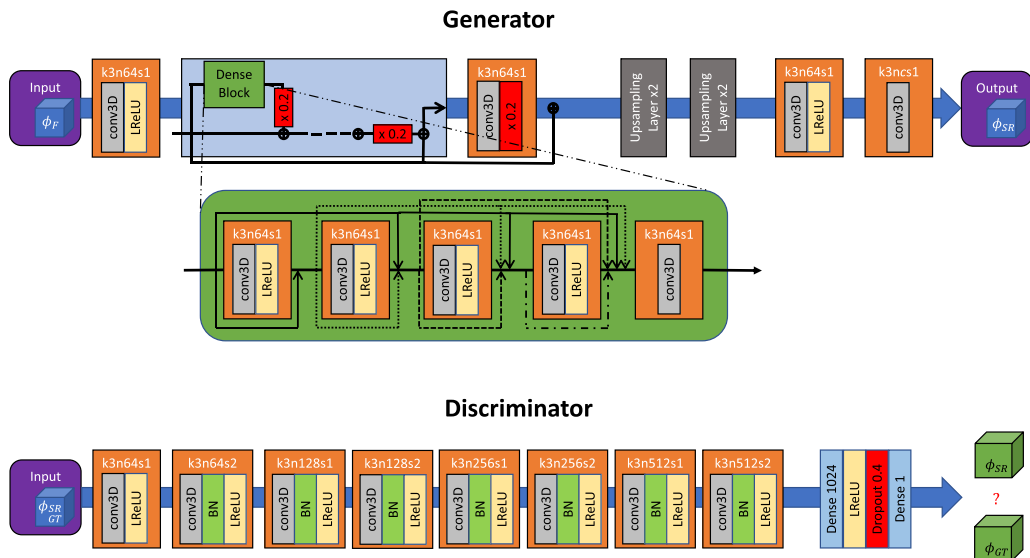
Generator

Discriminator

FIG. 1. Top: The generator structure employed for CNN and PIESRGAN-based models. Bottom: The discriminator structure employed for PIESRGAN models. In each, ϕ_F is the filtered input field, ϕ_{SR} is the super-resolved field, and ϕ_{GT} is the ground-truth (DNS) field. Each convolutional block contains kernels of size k, n filter maps, and s strides along each spatial dimension of the convolutional layer.

analyses), the filtered input field is rescaled to match the Δ/η ratio used for training. This rescaling procedure guarantees a certain level of generalizability [21].

III. NEURAL NETWORK ARCHITECTURES AND TRAINING STRATEGY

Our networks are based on the architecture of Bode *et al.* [19,20,40,41]. Differing from previous work, our generator network uses additional up-/downsampling layers and dense layers, depicted in Fig. 1; this is inspired by the original ESRGAN architecture [16] and adapted for small-scale turbulence reconstruction [22]. Along with the definition of the two architectures, the training approaches are described next.

A. TSResNet

The TSResNet (Turbulent Super-Resolution Residual Network) is a supervised CNN-type model. It uses only the generator shown in Fig. 1 without the discriminator. The TSResNet is adapted from the original SRResNet [42], which can capture small-scale features when skip connections are included. The generator relies on three-dimensional convolutional layers with Leaky Rectified Linear Units (LReLUs) as activation functions, as they have higher computational efficiency compared to standard rectified linear units (ReLUs) [33]. The Residual-in-Residual Dense Blocks (RRDBs) include residual connections, a set of densely connected-layer blocks (three in the present work), and a residual-in-residual structure [16]. Together, these enable super-resolved data to be generated through a very deep network. This capability is essential to facilitate learning complex transformations and is currently one of the state-of-the-art features of super-resolution networks [43]. Two upsampling layers increase the input spatial resolution by a factor of four in each spatial dimension. Each one doubles the dimensions of the input by nearest-neighbor interpolation (replicating adjacent grid points), followed by a convolution layer to improve the approximation. The upsampling branch (the gray section in Fig. 1) is later modified in Sec. V B, where additional upsampling layers are added. The total number of trainable parameters in the TSResNet is approximately 18 million.

The perceptual loss originally proposed by Wang *et al.* [16] for image reconstruction has been found to be less suitable for turbulence modeling [12] and is thus replaced with constraints derived from the continuity equation. The loss function used for the TSResNet training process ($\mathcal{L}_{\text{CNN}}$) is a combination of pixel loss (L_{pixel}), pixel gradient loss (L_{gradient}), and continuity loss ($L_{\text{continuity}}$),

$$
\begin{aligned}
\mathcal{L}_{\text{CNN}} &= \beta_1\, L_{\text{pixel}} + \beta_2\, L_{\text{gradient}} + \beta_3\, L_{\text{continuity}}, \\
L_{\text{pixel}} &= \text{MSE}(\phi_{\text{SR}}, \phi_{\text{DNS}}), \\
L_{\text{gradient}} &= \text{MSE}(\nabla\phi_{\text{SR}}, \nabla\phi_{\text{DNS}}), \\
L_{\text{continuity}} &= \text{MSE}(\nabla \cdot \phi_{\text{SR}}, 0),
\end{aligned}
\tag{6}
$$

where the coefficients $\beta = [0.89, 0.06, 0.05]$ were previously selected by Bode *et al.* [19] though hyperparameter tuning, ϕ_{SR} is the super-resolved field, and ϕ_{DNS} is the DNS field. The mean-squared error (MSE) is computed between the reconstructed and ground-truth fields and is applied separately to all elements when tensor quantities are considered. The L_{pixel} loss function is inherently dimensionless, owing to the prior normalization of both the input and output fields. Likewise, the L_{gradient} loss function is normalized using the Kolmogorov length scale η, as the gradients are small-scale quantities. Consequently, the $L_{\text{continuity}}$ function loss as the sum of normalized gradients is also dimensionless. This normalization helps to ensure that the loss functions are not affected by the choice of the grid spacing or the velocity magnitude, allowing for more general and scalable applicability of the nondimensional β parameters across similar configurations.

The network was initialized with random weights to ensure unbiased predictions. Training occurs for a sufficient number of epochs after which the loss function and statistics computed on the reconstructed field do not change substantially.

B. PIESRGAN

In contrast, the PIESRGAN model uses the full GAN architecture, i.e., generator and discriminator, in adversarial training. The generator has the same architecture as the TSResNet described previously. The discriminator (Fig. 1) is a deep deconvolutional architecture of fully connected layers with binary classification output, differing from the original ESRGAN discriminator by the introduction of a dropout layer to prevent overfitting. The total number of trainable parameters of the discriminator is approximately 19 million.

The generator's loss function is largely unchanged compared to the TSResNet but now includes the contribution of the discriminator loss ($L_{\text{adversarial}}$ [44]). The generator loss function is $\mathcal{L}_{\text{GAN}} = \beta_1\, L_{\text{pixel}} + \beta_2\, L_{\text{gradient}} + \beta_3\, L_{\text{continuity}} + \beta_4\, L_{\text{adversarial}}$, where

$$
\begin{aligned}
L_{\text{adversarial}} = &-\mathbb{E}[\log(\sigma(D(G(\phi_{\text{SR}})) - \mathbb{E}[D(\phi_{\text{DNS}})]))] \\
&- \mathbb{E}[\log(1 - \sigma(D(\phi_{\text{DNS}}) - \mathbb{E}[D(G(\phi_{\text{SR}}))]))],
\end{aligned}
\tag{7}
$$

$\mathbb{E}[\cdot]$ is the expectation operator, $\sigma(\cdot)$ is the sigmoid function, and $D(\phi_{\text{DNS}})$ and $G(\phi_{\text{SR}})$ are the discriminator and generator outputs. The first term of the adversarial loss function encourages the discriminator to correctly classify HR fields as real, while the second term encourages the generator to produce SR fields that can fool the discriminator into classifying them as HR. If ϕ_{DNS} is not provided during the training (unsupervised training; Sec. V), $L_{\text{adversarial}}$ is reduced to $L_{\text{adversarial}} = -\mathbb{E}[\log(\sigma(D(G(\phi_{\text{SR}}))))]$. The weights $\beta = [0.89, 0.06, 0.05, 6 \times 10^{-5}]$ were chosen through hyperparameter tuning, such that the absolute value of each term in $\mathcal{L}_{\text{GAN}}$ is of the same order. It is important to note that these β parameters may not be universally applicable and could be case-dependent. While those parameters have proven effective for HIT configurations, their optimality may vary for different flow configurations for which further hyperparameter tuning is recommended. The loss function, used to train the discriminator when ground truth DNS fields are

TABLE II. List of model configurations, training strategies, and training/testing filter kernels and sizes. The filter kernels "B," "G," and "S" abbreviate box, Gaussian, and spectrally sharp kernels, respectively. The $\wedge$ indicates that the combined set of the input fields is employed in the training, and the # indicates the variable upsampling factor.

Model	Training strategy	Input training–testing fields Filter kernels ($\overline{\Delta}/\Delta_{\mathrm{DNS}}$)	Section
Influence of adversarial training on in-sample predictions			
TSResNet (spectral kernel)	Supervised	S (4)–S (4)	Sec. IV
TSResNet (box kernel)	Supervised	B (4)–B (4)	Sec. IV
PIESRGAN (spectral kernel)	Semisupervised	S (4)–S (4)	Sec. IV
PIESRGAN (box kernel)	Semisupervised	B (4)–B (4)	Sec. IV
Influence of adversarial training on out-of-sample filters			
PIESRGAN (multiple-kernel-trained)	Semisupervised	B $\wedge$ S (4)–G (4)	Sec. V A
PIESRGAN (partially unsupervised)	Semisupervised and unsupervised	B $\wedge$ S $\wedge$ LES (4)–G (4)	Sec. V A
TSResNet (fixed upsampling)	Supervised	B (4)–B (8)	Sec. V B
PIESRGAN (fixed upsampling)	Semisupervised	B (4)–B (8)	Sec. V B
PIESRGAN (*x*#)	Semisupervised and dynamic upsampling	B (2 $\wedge$ 4 $\wedge$ 8)–B (#)	Sec. V B
Influence of adversarial training on out-of-sample predictions (higher Re numbers)			
TSResNet	Supervised (*Re90*)	B (4)–B (4)	Sec. VI
PIESRGAN	Semisupervised (*Re90*)	B (4)–B (4)	Sec. VI

available, is given by

$$
\begin{aligned}
\mathcal{L}_{\mathrm{DISC}} = {} & \mathbb{E}[\log(\sigma(D(\phi_{\mathrm{DNS}}) - \mathbb{E}[D(G(\phi_{\mathrm{SR}}))]))] \\
& + \mathbb{E}[\log(1 - \sigma(D(G(\phi_{\mathrm{SR}})) - \mathbb{E}[D(\phi_{\mathrm{DNS}})]))],
\end{aligned}
\tag{8}
$$

which is based on the relativistic average GAN loss function proposed by Jolicoeur-Martineau [44].

The training of the GAN architecture is challenging, as the generator and discriminator networks are trained to compete against each other. Finding a convergence point is one of the main challenges of GAN training, and training can suffer oscillations and destabilization in the model's trainable parameters [13]. To cope with these, the generator is pretrained in a fully supervised manner (like the TSResNet) by using the original generator loss function proposed in Eq. (6). Subsequently, the pretrained generator is used to initialize the GAN training. Recent findings demonstrated that the choice of the initial learning rate for both networks is essential for the local convergence of the GAN training, and recommendations from recent literature have been adopted [22,29].

C. Training strategy

Table II lists the architecture, training strategy, and training/testing fields for the specific trained models used in subsequent sections. In the supervised and semisupervised approaches, both LR and DNS field pairs are accessible, and the network utilizes this paired data set in an attempt to learn the mapping function between LR fields and to their corresponding ground-truth counterparts. This methodology is viable only when the data set contains paired data, i.e., both F-DNS and DNS data. In the unsupervised approach, only LR fields (LES data) are available without the corresponding DNS fields, thus learning relies solely on the feedback provided by the discriminator and the physics-informed loss function. The partially supervised approach is a combination of the two strategies, in which F-DNS/DNS pairs and true LES data as well as the discriminator feedback are employed for training.
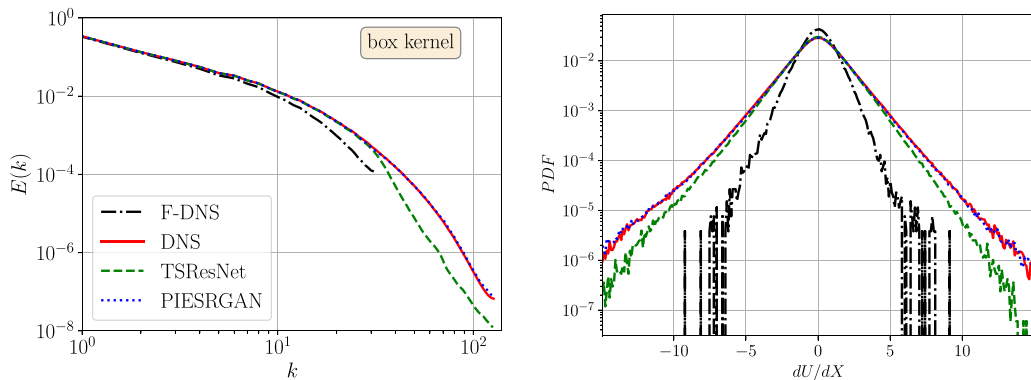
FIG. 2. In-sample (*Re90*) TKE spectra (left) and PDF of the normalized velocity gradient (right) for box-filtered training/testing fields.

The TSResNet models were trained for roughly 200 epochs, after which no additional reconstruction improvement was observed, using the *Re90* data set and Eq. (6) as the loss function. Statistical convergence required 37 wall-time hours using four NVIDIA Tesla V100 32 GB GPUs. The trained TSResNet model parameters were used to initialize the GAN generators. No GAN discriminator pretraining was applied, as this step has recently been demonstrated to have minimal influence [22]. The adversarial training required an additional 11 wall-time hours using the same hardware, i.e., roughly 30% more than the TSResNet training time, over an additional 40 epochs. To counteract the influence of the additional training time performed during the GAN training, the TSResNet models were trained for an additional 40 epochs. However, this additional training time did not impact the final reconstructed field and can be considered superfluous.

IV. EFFECTS OF ADVERSARIAL TRAINING ON IN-SAMPLE TURBULENCE RECONSTRUCTION

Both models are tested on in-sample data, statistically matched with the training data, to assess their reconstruction capability. This evaluates the models' performance under ideal conditions. Figures 2 and 3 compare TKE spectra and the probability density function (PDF) of normalized velocity gradients of the filtered DNS (F-DNS), full-mesh (unfiltered) DNS, TSResNet upsampled fields, and PIESRGAN upsampled fields. The SR models are trained to produce a statistically
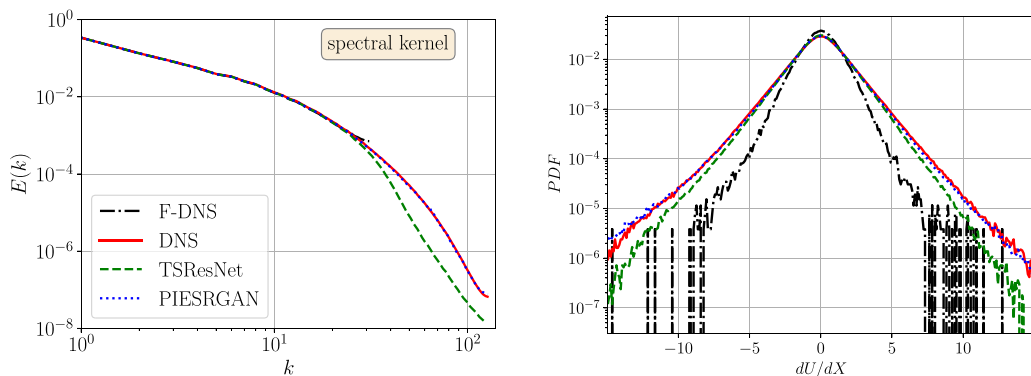

FIG. 3. In-sample (*Re90*) TKE spectra (left) and PDF of the normalized velocity gradient (right) for spectrally sharp filtered training/testing fields.
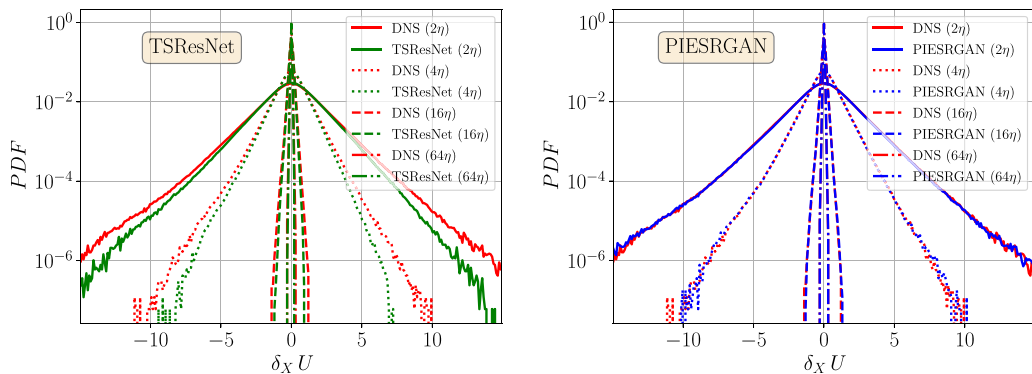
FIG. 4. PDFs of the normalized velocity increments for different separation distances for box-filtered DNS fields, TSResNet reconstructions (left), and PIESRGAN reconstructions (right).

correct SR field with DNS-like quality. The models are trained and evaluated for the *Re90* case using either the box filter (Fig. 2) or spectrally sharp filter (Fig. 3). The primary difference in the training data is the box filter's artificial attenuation of the near-filter-scale F-DNS fields, which the spectrally sharp filter avoids. The TSResNet recovers the TKE of the resolved scales ($k \gtrsim 32$) but fails to correctly predict the SFS energy distribution for either filter. Conversely, the PIESRGAN is more accurate, deviating from the DNS only for $k \gtrsim 122$ and energy values $E(k) \lesssim 10^{-7}$ for both filters. This behavior is quantitatively confirmed by the PDF of the normalized velocity gradient, where the TSResNet tends to underpredict the large gradients, while the PIESRGAN marginally deviates from the DNS solution. It is worth noting that the TSResNet and the PIESRGAN generator use the same network architecture, while the loss function defers only for the adversarial terms as defined in Sec. III B. The PIESRGAN's more accurate reconstruction is therefore due to its training, rather than the generator's network depth or loss-function differences. The PIESRGAN architecture improves the root-mean-squared error (RMSE) by 19% compared to the TSResNet for box-filtered training/testing fields and by 13% for spectrally sharp filtered training/testing fields.

PDFs of the normalized velocity increments $\delta_X U$ for different separation distances proportional to the Kolmogorov microscale η were considered to characterize turbulent structures at different scales, similar to Attili and Bisetti [45]. Figure 4 compares the DNS and reconstructed fields of velocity increments corresponding to distances of 2η (solid lines), 4η (dotted lines), 16η (dashed lines), and 64η (dash-dot lines). For small separation distances, the TSResNet captures only the most probable features well but underpredicts at the tails. For the largest increment (64η), the gap with the ground truth is less evident. These TSResNet reconstructions are perceptually smoother and blurred compared to the DNS fields, as a consequence of the poor reconstruction of the large normalized velocity increments. Conversely, the PIESRGAN predictions almost exactly overlap with the DNS for all increments considered. This alignment underscores that the structures recovered by PIESRGAN are coherent with the structures that were removed during the initial filtering operation. This is due to the discriminator's nonlinear feedback to the generator (due to the discriminator's nonlinear activation functions), which boosts the generator's accuracy for the subfilter scales. This means that the GAN can reconstruct complex, nonlinear relationships between low-resolution inputs and the reconstructed fields, independent of the filter kernel applied to the HR fields. The adversarial training, therefore, improves the in-sample reconstruction capabilities compared to only using supervised learning, even with the same generator architecture.

Additionally, the SR architecture's *a priori* SFS reconstruction accuracy and its effect on derived quantities are important considerations and a prerequisite for a potential *a posteriori* deployment in
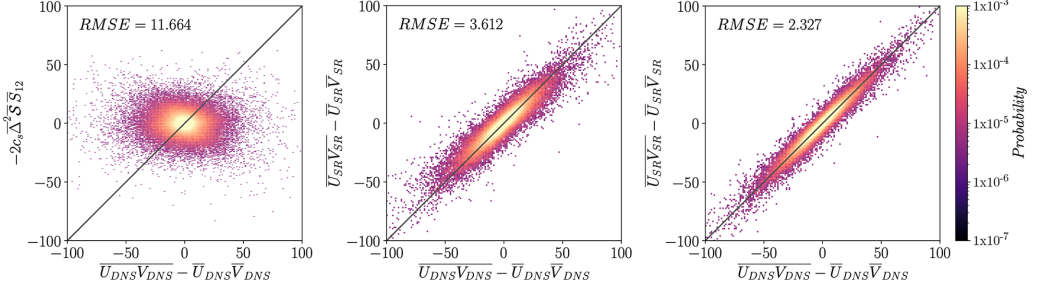
FIG. 5. Joint PDFs of the box-filtered DNS τ_{12}^{SFS} with the dynamic Smagorinsky-modeled field (left), TSResNet reconstruction (center), and PIESRGAN reconstruction (right).

LES. The LES-filtered momentum equation,

$$\frac{\partial \overline{u}_j}{\partial t} + \overline{u}_i \frac{\partial \overline{u}_j}{\partial x_i} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j} + \nu \frac{\partial^2 \overline{u}_j}{\partial x_i \partial x_i} - \frac{\partial \tau_{ij}^{\text{SFS}}}{\partial x_i}, \tag{9}$$

contains the anisotropic residual-stress tensor τ_{ij}^{SFS},

$$\tau_{ij}^{\text{SFS}} \equiv \overline{u_i u_j} - \overline{u}_i\,\overline{u}_j - \frac{2}{3}k^{\text{SFS}}\delta_{ij}, \tag{10}$$

where u_i are unfiltered velocity components obtained from either the DNS or the SR fields and $k^{\text{SFS}} \equiv (\overline{u_i u_i} - \overline{u}_i \overline{u}_i)/2$ is the kinetic energy of the subfilter scales [46]. In Eq. (10) the $\overline{\cdot}$ operator indicates filtering (either box or spectrally sharp filters) and downsampling proportional to the upsampling factor (in the present case, the factor is equal to four). Comparisons are made to the widely used dynamic Smagorinsky model [38,39],

$$\tau_{ij}^{\text{SFS}} = -2c_s\overline{\Delta}^2\,|\overline{S}|\overline{S}_{ij}, \text{ with } |\overline{S}| = (2\overline{S}_{ij}\overline{S}_{ij})^{1/2}, \tag{11}$$

where $\overline{S}_{ij}$ denotes the filtered strain-rate tensor, $\overline{\Delta}$ is the filter width, and c_s is the dynamically computed Smagorinsky constant, averaged along the homogeneous directions [39].

Figure 5 shows joint PDFs (jPDFs) of the box-filtered DNS τ_{12}^{SFS} and the dynamic-Smagorinsky (left), TSResNet-reconstructed (center), and PIESRGAN-reconstructed (right) τ_{12}^{SFS}. The PIESRGAN-reconstructed field is statistically more similar to the filtered DNS than the TSResNet-reconstructed and dynamic-Smagorinsky-modeled fields, in order of decreasing fidelity. In general, each component of the PIESRGAN-reconstructed τ_{ij}^{SFS} has an average pointwise correlation, computed from the jPDF, with the DNS exceeding 90%. Similar results are obtained for spectrally filtered input fields. It is clear that adversarial training improves the *a priori* correlation of SR fields with ground-truth data for in-sample predictions.

Similarly, the SFS dissipation rate $\varepsilon^{\text{SFS}} = \overline{S}_{ij}\,\tau_{ij}^{\text{SFS}}$ is highlighted because it is a crucial property of SFS models [47]. Figure 6 shows the jPDFs of the box-filtered DNS SFS dissipation rate and those of the dynamic-Smagorinsky model (left), the TSResNet-reconstructed field (center), and the PIESRGAN-reconstructed field (right) ε^{SFS}. Here τ_{ij}^{SFS} is determined using Eq. (11) for the dynamic Smagorinsky model and Eq. (10) for the SR models and DNS. The dynamic Smagorinsky model demonstrates its dissipative nature, evident in the exclusive occurrence of positive values for $\varepsilon_{\text{SMAG}}^{\text{SFS}}$. Locally, there are strong deviations from the SFS dissipation rate of the DNS. The data-driven models, on the other hand, predict dissipation rates that are in superior local agreement with the DNS. The TSResNet-reconstructed $\varepsilon_{\text{SR}}^{\text{SFS}}$ distribution is slightly bulky around the diagonal, while the PIESRGAN-reconstructed $\varepsilon_{\text{SR}}^{\text{SFS}}$ is closer to the correct range of the *a priori* local SFS dissipation. The average SFS dissipation rate $\langle \varepsilon^{SFS} \rangle$ computed using the TSResNet is roughly 50% higher compared with that computed using the PIESRGAN network. The mean SFS dissipation predicted by the PIESRGAN is within 2% of the DNS value.
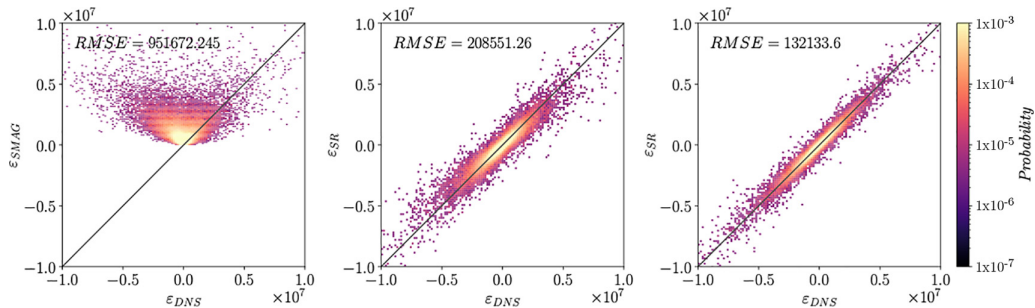
FIG. 6. Joint PDFs of the box-filtered DNS ε^{SFS} with the dynamic Smagorinsky-modeled field (left), TSResNet reconstruction (center), and PIESRGAN reconstruction (right).

V. ACCURACY OF SR MODELS FOR OUT-OF-SAMPLE FILTERS

Recent work has highlighted practical limitations of data-driven deconvolution methods arising primarily when the application differs substantially from the training data [13,26,48]. In the previous section, the PIESRGAN successfully reconstructed in-sample filtered fields ϕ_F and approximately inverted the operator $\mathcal{G}^{-1}$. However, training SR models using only F-DNS data with corresponding full-resolution fields does not necessarily guarantee accurate upscaling of real-world fields having potentially different turbulence statistics [49]. The reason could be that for *a posteriori* LES modeling, the interactions between grid spacing, numerical accuracy, and modeling assumptions might have to be considered. In the context of experimental analysis, the filter kernel and width implied by optical and camera systems are unknown. To understand the influence of these parameters, we now consider out-of-sample filter kernels and filter widths separately.

A. Different filter kernels for training and testing

The trained SR models are now tested for F-DNS input fields using out-of-sample filter kernels but the same filter size. This emulates applications of trained models on fields for which the implicit filter operation is unknown, such as the use of experimental data at insufficiently high resolution.

Figure 7 compares the TKE spectra and the PDF of normalized velocity gradients for the networks that were trained individually on box-filtered or spectrally sharp filtered fields exclusively and applied to Gaussian-filtered input data. The exclusively box-filter-trained SR models consistently overestimate the SFS kinetic energy, which results in inaccurate predictions at the high wave
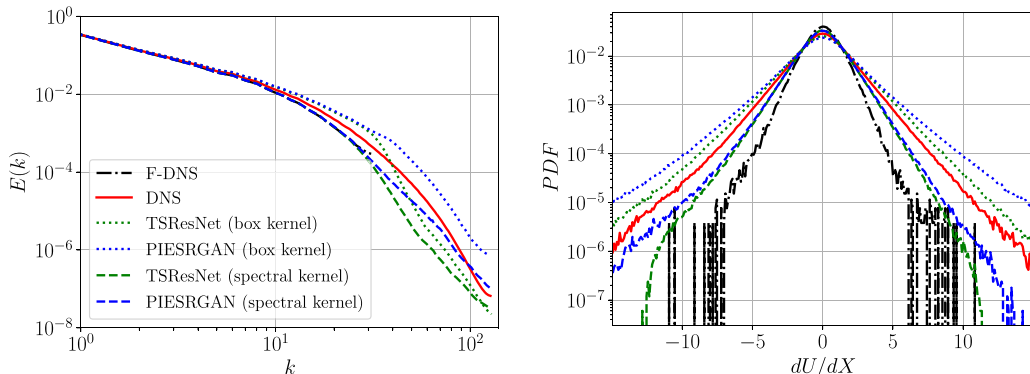

FIG. 7. TKE spectra (left) and PDF of the normalized velocity gradient (right) for models trained either for box- or spectrally sharp-filtered data and tested on Gaussian-filtered data.

numbers and overpredicted energy at lower wave numbers than the filter cutoff. Conversely, for models exclusively trained on spectrally sharp filtered data, the opposite behavior is observed when the network is applied to Gaussian-filtered data: both models underpredict the resolved and SFS kinetic energy, with the PIESRGAN model being slightly more accurate than the TSResNet model. Clearly, both SR models fail to extrapolate to filter kernels not seen in training.

One possible approach to address the reduced out-of-sample filter invertibility is to combine an SR architecture with an algebraic approximate deconvolution method [50]. This would first use the ADM to recover the resolved velocity fluctuations, then a PIESRGAN trained for spectrally sharp filtered fields would extrapolate the subfilter scales. Still, it is important to note that the ADM-deconvoluted field would still differ from an ideal, spectrally sharp filtered field, especially near the cutoff scale [11,51]. Thus, the performance of the spectral-trained GAN could suffer from its tendency to invert a known filter. A combination of an SR data-driven model and an algebraic ADM could partially minimize the GAN's over/underestimation, but it would likely not completely solve the issue.

Another potential solution leverages the GAN's flexibility to train both in a semisupervised and unsupervised mode, relying on the underlying "structure" of the data for patterns. This approach will be described subsequently. Both models were trained on a mixture of box-filtered and spectrally sharp filtered data in an otherwise consistent manner with the training strategy outlined previously (equal number of subboxes per filter kernel, randomly selected during the training). This approach is henceforth referred to as *multiple-kernel-trained*. The discriminator is then trained to distinguish between generated high-resolution fields, upsampled from either box-filtered or spectrally sharp filtered DNS fields, and the corresponding ground-truth fields. Validating for in-sample F-DNS fields (i.e., obtained with the same filter kernels used for training), the reconstruction performance is comparable to that obtained in Sec. IV. Similar to the previous analysis, the PIESRGAN demonstrates better reconstruction capability than the TSResNet; hence we omit the TKE spectra and PDF of the normalized velocity gradient for brevity. Both models are capable of recognizing and learning both filtering operations that were applied to generate low-resolution training data, suggesting that this ability is not exclusively derived from adversarial training.

When testing for out-of-sample Gaussian-filtered input data (not included in the training data), the multiple-kernel-trained PIESRGAN model again suffers a performance drop. Figure 8 plots centerline slices of the normalized velocity magnitude and its absolute error obtained for the PIESRGAN using multiple-filter-kernel training. It is evident that the velocity field reconstructed by the model exhibits distortions compared to the ground-truth data. Notably, the error is especially pronounced for small-scale structures. Figure 9 shows TKE spectra and PDFs of normalized velocity gradients for the multiple-kernel-trained PIESRGAN model, showing that the reconstructed field is in better agreement with the DNS than the single-filter-trained model, though artifacts remain in the high-frequency subfilter scales. The velocity-gradient PDF shows similar trends for the multiple-kernel-trained PIESRGAN, with consistent overprediction of the small-scale shear strain. Training for different filter kernels can mitigate filter-inversion inadequacies, but does not alleviate them altogether.

While the discriminator does not directly influence the generator's ability to distinguish between filter kernels, adversarial training can still improve the accuracy of the reconstructed fields. This is an advantage of the GAN-based architecture. Inspired by Bode *et al.* [19], though differing in purpose and methodology, we now include both labeled (corresponding DNS and F-DNS fields)—as in the previous approach—and unlabeled data (computed LES fields, either with no SFS model or the dynamic Smagorinsky model, as described in Sec. III) in the training process (referred to as the LES-training step). Labeled data help the discriminator differentiate between real and fake samples, while unlabeled data improve the generator's ability to upsample realistic fields. The discriminator is trained in a supervised manner using only generated and ground-truth fields, and its weights are not updated in the LES-training step, while the generator continues to be trained on randomly sampled LES data. The resulting super-resolved LES fields are then evaluated by the trained discriminator.
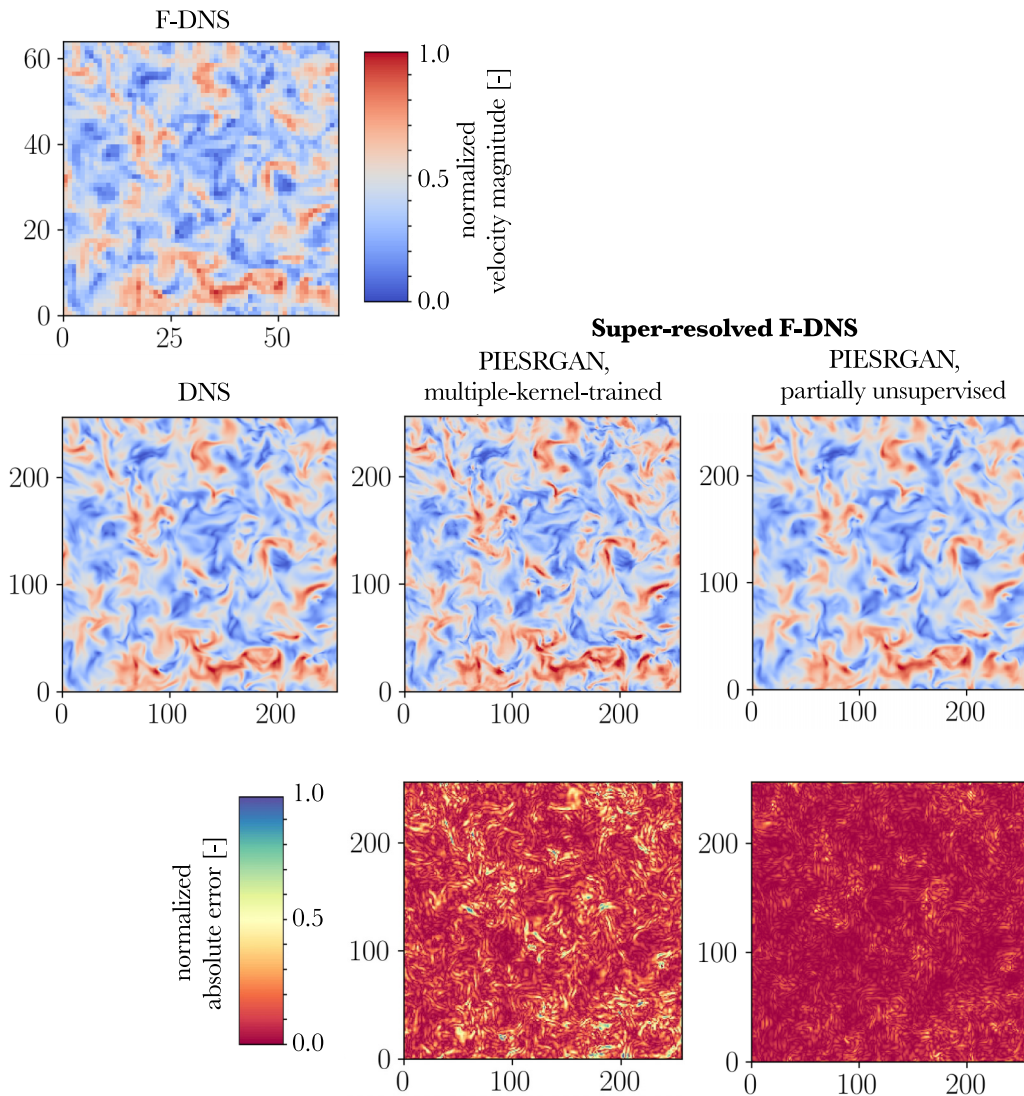
FIG. 8. 2D slices of instantaneous normalized velocity magnitude and the absolute error of SR fields versus DNS. The normalized absolute error is $\hat{E} = E / \max(E_{DNS,SR_1}, E_{DNS,SR_2})$, where $E = |u_{DNS} - u_{SR_{1,2}}|$, and the subscripts 1 and 2 indicate the multiple-filter-kernel and semisupervised training approaches, respectively.

Since the discriminator is not trained further, the generator loss was reduced to

$$\mathcal{L}_{GAN_{unsupervised}} = \beta_3 L_{continuity} + \beta_4 L_{adversarial}, \tag{12}$$

where $\beta_3 = 0.05$ and $\beta_4 = 6 \times 10^{-5}$. This is referred to as the partially unsupervised approach.

The right column of Fig. 8 demonstrates that the PIESRGAN model, trained in a partially unsupervised manner, precisely reconstructs small-scale structures absent from the (out-of-sample) input data, with significantly lower normalized absolute error. Thus, the use of partially unsupervised training significantly reduces the overprediction of small features. In terms of RMSE, the reconstructed field from the PIESRGAN trained with multiple kernels is 67% higher compared to that computed using the partially unsupervised-trained PIESRGAN. Figure 9 illustrates decreased
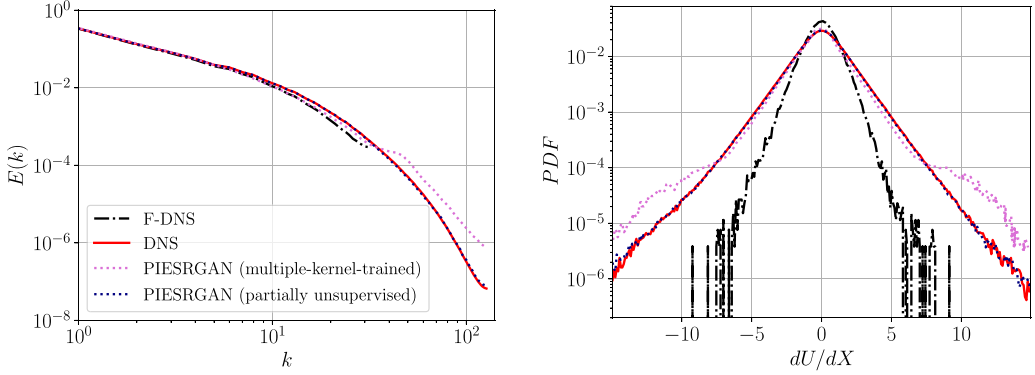
FIG. 9. TKE spectra (left) and normalized velocity gradient PDFs (right) for PIESRGAN models trained for multiple filter kernels without and with partially unsupervised step (LES-training step). Testing results are shown for Gaussian-filtered fields.

overprediction of the SFS kinetic energy and better reconstruction of the normalized velocity gradient PDF when the partially unsupervised training is employed. The significant improvements at both large and small scales indicate that the unsupervised training step (LES-training step), possible only with GAN-based models, helps to generalize the model to diverse input fields.

It is worth noting that using computed LES fields or F-DNS fields as input is somehow equivalent to including random fields, as typically used in GAN applications, without incurring the risk of introducing spurious non-physical phenomena. This is important because the large scales need to be consistent between LR and HR fields. This choice enables the generator to focus on refining and augmenting existing subfilter-scale structures only, providing a guided generation process. Acknowledging the notorious instability and convergence difficulties in GAN training, we prefer this LR field-based solution over incorporating random vectors/fields.

To isolate the effect the discriminator exerts, we disable either the adversarial or continuity loss in Eq. 12. With this, the generator is trained in a fully unsupervised manner, relying solely on feedback from the continuity equation (first option) or solely on feedback from the discriminator network (second option). Figure 10 compares the possible training options showing joint PDFs of the Gaussian-filtered DNS τ_{12}^{SFS} [Eq. (10)] using only $L_{\mathrm{continuity}}$ loss (left), only $L_{\mathrm{adversarial}}$ loss (center), and both $L_{\mathrm{continuity}}$ and $L_{\mathrm{adversarial}}$ losses (right) during the unsupervised training phase. The PIESRGAN-reconstructed field using only the $L_{\mathrm{adversarial}}$ loss during unsupervised training is statistically similar to that obtained when both losses are employed. Conversely, the reconstructed field obtained using only the $L_{\mathrm{continuity}}$ depicts a bulkier distribution around the diagonal and a
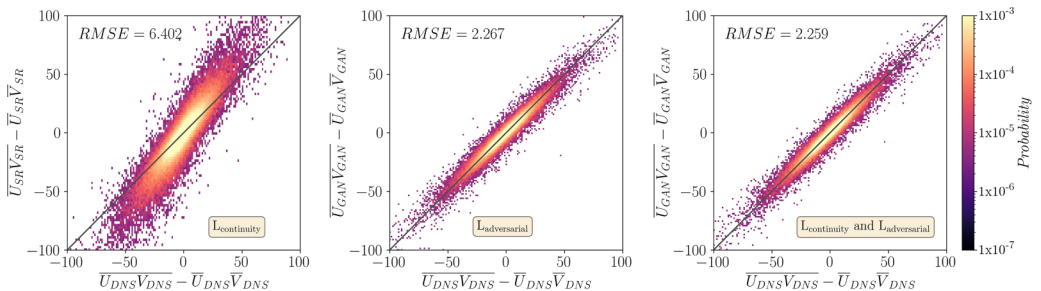

FIG. 10. Joint PDFs of the Gaussian-filtered DNS τ_{12}^{SFS} using only $L_{\mathrm{continuity}}$ loss (left), only $L_{\mathrm{adversarial}}$ loss (center), and both $L_{\mathrm{continuity}}$ and $L_{\mathrm{adversarial}}$ losses (right) during the unsupervised training phase.
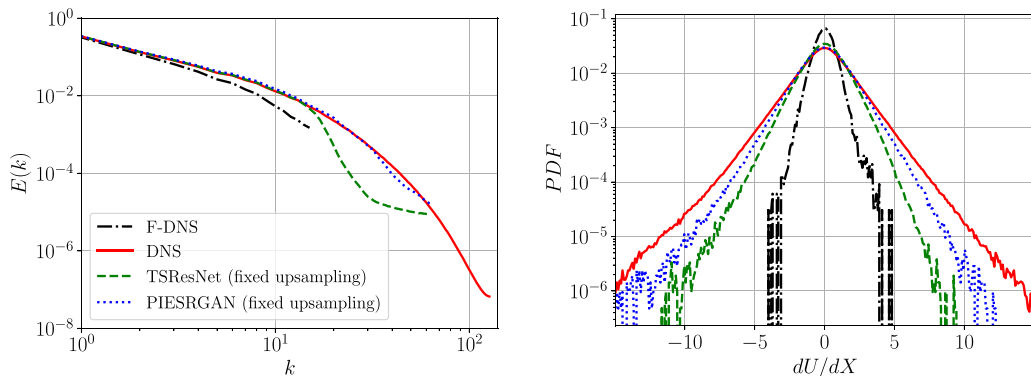
FIG. 11. TKE spectra (left) and the PDF of the normalized velocity gradients (right) when both models are trained with fields filtered with a box filter with a size of four and tested on a field filtered with a box filter with a size of eight.

predominant shift, resulting in a consistent overestimation of the magnitude of the Reynolds stress. While the reconstruction improvement using only $L_{adversarial}$ is comparable to that shown in Fig. 9, the training process became more unstable and ultimately resulted in GAN collapse. To address this issue, the batch size and initial learning rate were decreased (the batch size was divided by half and the initial learning rate decreased by an order of magnitude) to improve the training stability [22]. Thus, the continuity loss serves to stabilize and accelerate the training process and does not seem to meaningfully drive reconstruction performance, but further investigations are needed.

B. Different filter sizes for training and testing

We now explore the ability of SR models to reconstruct fields for different filter sizes than from those used in training. SR training typically uses fixed upsampling factors, which can result in blurry fields and/or artifacts for upsampling factors not included in the training data. This is especially critical for flow conditions and/or geometries requiring different *local* filter ratios (e.g., near-wall resolution for PIV or stretched grids for LES computations). For this, a fixed SR upsampling factor can cause over- or under-sampling, leading to reduced reconstruction capabilities.

To investigate this, SR models trained to upsample data for an LES-to-DNS-mesh ratio $\overline{\Delta}_4 = \overline{\Delta}/\Delta_{DNS} = 4$, where $\overline{\Delta}$ is the effective LES mesh resolution and Δ_{DNS} is the DNS mesh resolution, are tested for $\times 4$ upsampling of $\overline{\Delta}_8$ input data—that is, upsampling $\overline{\Delta}_8$ data to an effective LES resolution of $\overline{\Delta}_2$. All tests use box-filtered data.

Figure 11 shows TKE spectra and normalized velocity-gradient PDFs for the previously supervised box-trained TSResNet and PIESRGAN models applied to box-coarser *Re90* input data. The PIESRGAN accurately recovers the TKE spectrum over the upsampled range, even though it does not fully recover the low-probability tails of the velocity-gradient PDF. The TSResNet model does not fare as well, with significant deviations from the DNS spectrum throughout the upsampled range. Despite these imperfections, it is clear that the PIESRGAN exhibits an improved tendency to successfully upsample across different scale ranges than those used for training, at least within the self-similar scales of statistically stationary HIT. Of course, this does not imply equal success for anisotropic flows.

Multiple upsampling branches can be incorporated into the generator network to determine the ideal upsampling ratio from the input data, as applied by Lim *et al.* [52] for image-processing tasks. We now modify the generator architecture (Fig. 1) for multiple upsampling branches, shown in Fig. 12. Three branches enabling different upsampling ratios are present.

(1) The first branch, for high-resolution inputs, contains one $\times 2$ upsampling layer.
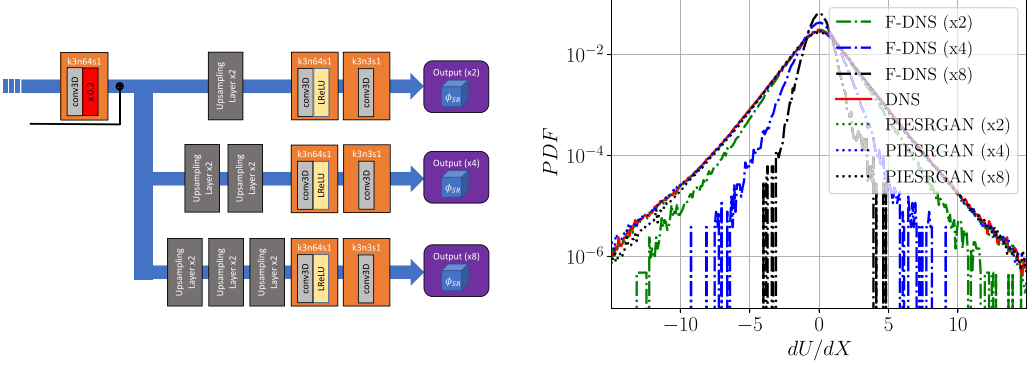
FIG. 12. Left: Sketch of the three upsampling branches in the modified GAN generator. Right: PDF of the normalized velocity gradient for PIESRGAN fields reconstructed using the most appropriate upsampling factor selected by the discriminator.

(2) The second branch, for intermediate-resolution inputs (used in Sec. IV), contains two upsampling layers for an overall ×4 upsampling ratio.

(3) The third branch, intended for low-resolution inputs, contains three upsampling layers for an overall ×8 upsampling ratio.

Multiple branch employment is promising for image processing [52], though the generator needs prior information on the input size, resulting in the network being able to handle only predetermined, known input sizes. Small upsampling factors typically result in adequate reconstruction, if incompletely resolved, with larger structures typically accurately captured. Conversely, over-upsampling can lead to significant spurious artifacts [53].

Instead, we use the discriminator to select the appropriate upsampling branch. The GAN is trained for box-filtered fields of different filter sizes ($\overline{\Delta}_2$, $\overline{\Delta}_4$, and $\overline{\Delta}_8$). During training, the correct upsampling branch is enabled based on the known filter size, and unused branches are deactivated. During testing, the generator upsamples the input field along all three branches, producing ×2, ×4, and ×8 fields, which the trained discriminator then uses to calculate an adversarial loss [using Eq. (8) with $\phi_{\mathrm{DNS}} = 0$]. The upsampling factor corresponding to the lowest adversarial loss is chosen. Thus, the discriminator learns to recognize high-quality fields during training and can provide highly nonlinear feedback to the generator on the quality of the generated field.

Table III lists a "quality score"

$$Q = \mathbb{E}[\log(\sigma(D(G(\phi_{\mathrm{SR}}))))], \tag{13}$$

related to the discriminator loss, for $\overline{\Delta}_2$, $\overline{\Delta}_4$, $\overline{\Delta}_8$, and $\overline{\Delta}_{16}$ input data upsampled along the ×2, ×4, and ×8 branches. Higher values indicate higher confidence (by the discriminator) in the quality of

TABLE III. Quality score Q for different input filter-to-DNS-grid ratios and upsampling branches.

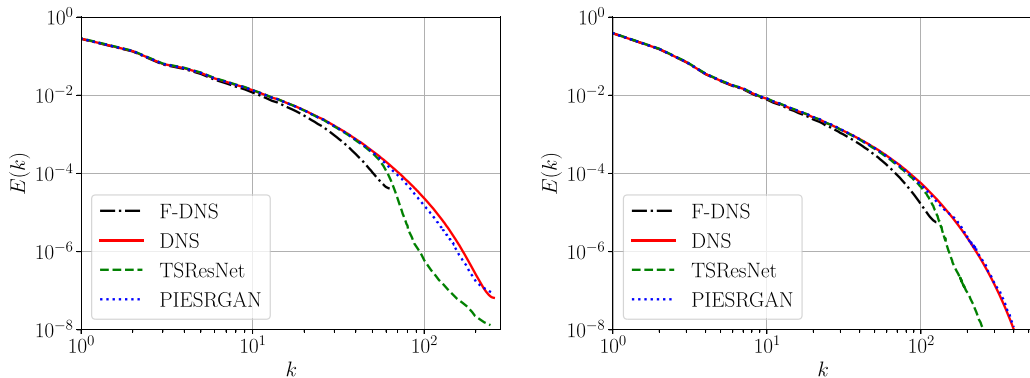| Input	Upsampling factor		
	×2	×4	×8
$\overline{\Delta}_2$	0.93	0.46	0.08
$\overline{\Delta}_4$	0.73	0.91	0.23
$\overline{\Delta}_8$	0.53	0.69	0.84
$\overline{\Delta}_{16}$	0.17	0.34	0.59

FIG. 13. TKE spectra for *Re90*-trained models evaluated on higher-Reynolds-number *Re130* (left) and *Re350* (right) input data. Filtered DNS data were obtained using a $\overline{\Delta}_4$ box filter.

the upsampled field. The $\overline{\Delta}_{16}$ input data are out-of-sample. The branch providing the highest quality score is selected as the most-appropriate output for arbitrary input data.

Figure 12 shows normalized velocity-gradient PDFs for PIESRGAN-upsampled fields using the discriminator-selected upsampling branches. It is evident that the discriminator accurately selects the appropriate branch, enabling the modified PIESRGAN to accurately recover both the large scales and the tails of the PDFs. This adaptive upsampling strategy, utilizing the discriminator, is particularly effective in scenarios where the input filter size is not present in the training data. For example, in Table III, the $\times 8$ upsampling branch is correctly selected as the most appropriate available branch for the out-of-sample $\overline{\Delta}_{16}$ input data. However, it is important to note that this approach entails higher computational cost compared to using a single upsampling factor (roughly 50% higher), as more fields are employed. In conclusion, this dynamic upsampling feedback, which can be provided only in the context of a GAN-based architecture, is an effective approach to mitigating the fixed upsampling/filter size limitation.

VI. EXTRAPOLATION TO HIGHER REYNOLDS NUMBERS

The previous sections demonstrate that the PIESRGAN architecture can reconstruct unresolved fields more accurately than the TSResNet for in-sample flows. The discriminator, playing a crucial role in adversarial training, permits greater flexibility for out-of-sample conditions (e.g., for different filter kernels and/or upsampling ratios), though the model must also generalize well beyond the flow conditions used for training.

To test this generalization capability with equal filter kernel and size for both training and testing, thus under ideal conditions, the models trained for the lower $Re_\lambda = 90$ case (*Re90*) (Sec. IV) are applied to forced HIT for the higher $Re_\lambda = 130$ (*Re130*) and $Re_\lambda = 350$ (*Re350*) cases (Table I). Figure 13 compares the TKE spectra for these out-of-sample model applications to the unfiltered and box-filtered ($\overline{\Delta}_4$) DNS spectra. The $Re_\lambda = 90$-trained TSResNet model fails to correctly reconstruct the subfilter scales, similar to its performance in the previous in-sample analyses, while the $Re_\lambda = 90$-trained PIESRGAN model accurately extrapolates the subfilter scales at these higher Reynolds numbers, at least outside the dissipation range. The ability of the PIESRGAN model to extrapolate the small scales of anisotropic flows remains to be studied.

The architecture can capture the scale similarity of the turbulent motions and be applied at various Reynolds numbers on similar configurations. It is worth noting that the higher-Reynolds-number reconstruction (*Re350*) is slightly more accurate than the-lower-Reynolds-number reconstruction (*Re130*). This is due to the ratio Δ/η differing slightly from the one used for training (Sec. III). In both extrapolation investigations, the PIESRGAN reconstruction indicates a 17% reduction in
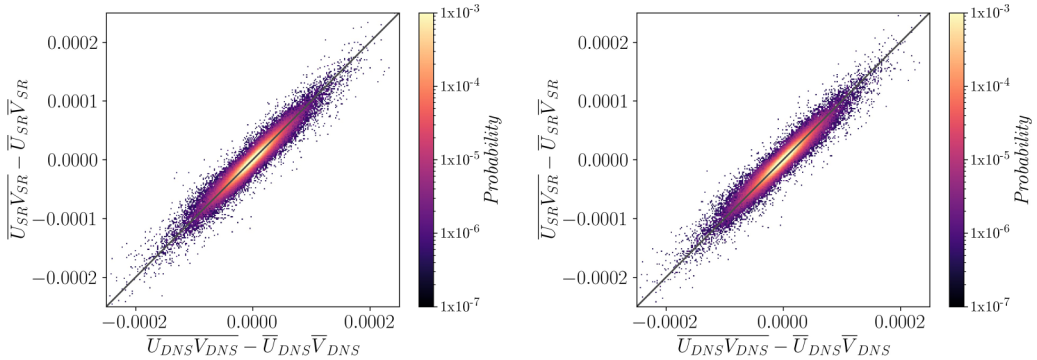
FIG. 14. Joint PDFs of the box-filtered τ_{12}^{SFS} with PIESRGAN reconstruction when applied to the *Re130* data set (left) and the *Re350* data set (right).

RMSE compared to the reconstruction achieved by TSResNet. The jPDFs of τ_{12}^{SFS} evaluated on the PIESRGAN-reconstructed field for*Re130* and *Re350* inputs are shown in Fig. 14. In both cases, the PIESRGAN model extrapolates accurately to higher Re, with remarkable alignment of the predicted SFS stress with the DNS data. In both cases, the cross-correlation of every SFS stress tensor component computed from the fields obtained from the PIESRGAN model exceeds 90%.

VII. CONCLUSION

We investigated the influence of adversarial training on super-resolution turbulence reconstruction. GAN-based models (PIESRGANs) are assessed against a standard, supervised CNN-based model (TSResNet). Two approaches are introduced that leverage the GAN model's adversarial training capabilities, enabled by its discriminator network, to enhance the generator network's accuracy and generalizability to out-of-sample inputs for different filter sizes and filter kernels. The GAN generator and TSResNet have identical model structures, hence their evaluation cost is the same despite the enhanced accuracy of the GAN model.

The two training approaches have first been evaluated for in-sample data (i.e., statistically similar to the training data). In TKE spectra and velocity-gradient PDFs, the PIESRGAN architecture more accurately reconstructs the full-resolution velocity fields compared to the TSResNet model. The TSResNet model accurately captures the large-scale features, though the PIESRGAN is more consistent throughout the range of scales and velocity increments.

Adversarial training, while computationally more expensive, is promising to enhance SR model reconstruction of small-scale turbulence. In *a priori* assessments of the SFS stress tensor, the SR models outperform the widely used dynamic Smagorinsky model, showing significantly improved statistical alignment with the "true" filtered-DNS SFS stresses. The usefulness of these findings will, however, need to be further verified through *a posteriori* tests. Of the SR models, the PIESRGAN-reconstructed SFS fields are 64% better aligned with the filtered-DNS SFS fields than those produced by the TSResNet. This is due entirely to the GAN discriminator's feedback to the generator; all other training was performed identically. The discriminator's semisupervised training enables it to capture more complex, nonlinear relationships between the low- and high-resolution fields than a standard, supervised training approach would permit.

Data-driven super-resolution methods are known to suffer significant performance drops when applied to substantially out-of-sample data. This capability is particularly important to upscale real-world fields. To test this, we investigated two non-consistent applications of the SR models to testing data sets: out-of-sample filter kernels and out-of-sample upscaling ratios. To assess the influence of the training filter kernel, box- and spectrally sharp-filtered SR models were applied to Gaussian-filtered DNS fields. The box/spectral-trained SR models consistently over/underestimated the

SFS kinetic energy when applied to Gaussian-filtered fields and produced poorly aligned jPDFs of τ_{12}^{SFS}. To mitigate this behavior, both architectures were trained for a random collection of fields containing both box- and spectral-filtered data. The ensuing models have slightly greater reconstruction capability than the single-filter-trained models, though both still had a consistent gap to the Gaussian-filtered DNS. The TSResNet was again inaccurate for the subfilter scales, while the PIESRGAN added artifacts at high wave numbers, with slight underprediction of filter-scale features.

To improve the GAN generator's ability to upsample unlabeled, low-resolution fields (i.e., without accompanying high-resolution fields), computed LES fields were provided as LR training inputs. This choice enabled the generator to refine and augment existing SFS structures only from input fields that possess physical meaning, ensuring an effective contribution to the generative process. The discriminator was employed in an unsupervised manner to provide feedback to the generator during training. By combining inputs with multiple kernels and computed LES data, the generator was trained to produce fields with correct high-frequency details emulating the DNS. This partially unsupervised approach permits the PIESRGAN architecture to handle diverse input fields more effectively, and the issue of consistent small-scale overprediction of energy density is alleviated.

The ability of SR models to upsample out-of-sample filter-to-DNS mesh ratios was also explored. When a single upsampling factor is used for training, the reconstructed field may exhibit blurriness and artifacts, leading to inaccurate super-resolved fields and decreased predictive capability. To overcome this issue, two additional upsampling branches were added to the generator, with the discriminator determining which branch to apply locally. This enables the GAN model to manage input fields of potentially unknown resolution. By selecting the upsampling factor that minimizes this adversarial loss, the generator most faithfully reconstructs a given input field, at least within the range of upsampling ratios provided during training. We find that training over different filter kernels, unlabeled data (e.g., true LES data), and multiple filter-to-DNS mesh ratios enables sufficient model robustness and reconstruction capabilities.

Finally, the ability of the two models to extrapolate to higher Reynolds numbers was tested. Using input fields from higher-Reynolds-number DNS, filtered to maintain the same $\overline{\Delta}/\eta$ ratio used for training, the PIESRGAN architecture more consistently reconstructed small-scale flow features than the TSResNet. This suggests that GAN-based models trained for lower Reynolds numbers can be successfully applied to slightly higher Reynolds numbers, though further testing for non-HIT flows (e.g., turbulent shear flows, turbulent reacting flows) is ongoing.

Adversarial training, although computationally more costly than standard supervised approaches, is overall successful in improving the reconstruction and extrapolative capabilities of the data-driven SR model. Leveraging the discriminator's feedback, a GAN-based model is able to generate more diverse and realistic samples during training that improve the architecture's performance and generalizability to unseen training data.

In order to enhance the reproducibility of this study and provide additional clarity on technical aspects of the architectures, access to our GIT repository will be provided upon request.
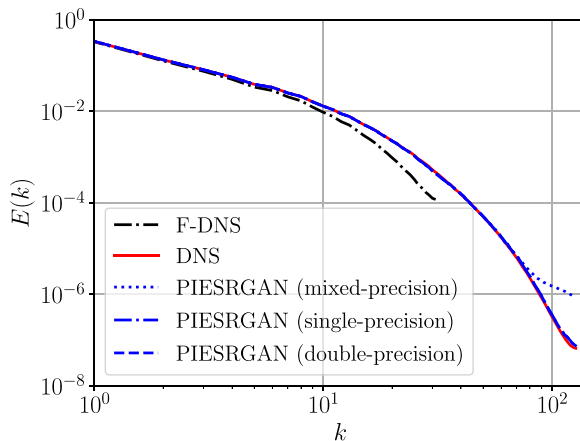
FIG. 15. Comparison of the TKE spectra for in-sample box-filtered training/testing fields when different numerical-precision training are adopted. Mixed-precision combines half-precision floating point (*FP16*) with single-precision floating point (*FP32*).

APPENDIX: INFLUENCE OF NUMERICAL PRECISION

"Mixed-precision" training has emerged as a powerful technique with significant implications for deep learning models. In conventional training, using 32-bit floating-point precision ("single precision," *FP32*) for storing model parameters incurs higher memory requirements. To address this limitation and accelerate model training, reduce memory usage, and improve inference times, mixed-precision training has gained popularity. It combines *FP16* (16-bit floating-point precision) and *FP32*, for different parts of the neural network computation. Its major drawback is that reduced numerical precision can considerably reduce the precision of the resulting model, which is particularly detrimental for highly multiscale problems [37]. Lower precision can compromise a trained model's parameters and increase the probability that the magnitude of the loss gradients could be in the same order as the floating-point roundoff error.

To evaluate the influence of numerical precision, Fig. 15 shows TKE spectra for in-sample, box-filtered, super-resolved fields from PIESRGAN models trained using mixed, single, and double precision (*FP64*). Mixed-precision training clearly leads to an accumulation of TKE at high wave numbers, while both single and double precision are visually comparable to the DNS at these high wave numbers. Because it is computationally less intensive, single precision is therefore chosen for the model-training results presented throughout the paper.

---

[1] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. **51**, 357 (2019).

[2] H. Wang, Z. Yang, B. Li, and S. Wang, Predicting the near-wall velocity of wall turbulence using a neural network for particle image velocimetry, Phys. Fluids **32**, 115105 (2020).

[3] K. Fukami, K. Fukagata, and K. Taira, Super-resolution analysis via machine learning: A survey for fluid flows, Theor. Comput. Fluid Dyn. **37**, 421 (2023).

[4] H. Kim, J. Kim, S. Won, and C. Lee, Unsupervised deep learning for super-resolution reconstruction of turbulence, J. Fluid Mech. **910**, A29 (2021).

[5] K. Fukami, K. Fukagata, and K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, J. Fluid Mech. **870**, 106 (2019).

[6] K. Fukami, K. Fukagata, and K. Taira, Assessment of supervised machine learning methods for fluid flows, Theor. Comput. Fluid Dyn. **34**, 497 (2020).

[7] K. Fukami, K. Fukagata, and K. Taira, Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows, J. Fluid Mech. **909**, A9 (2021).

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA* (IEEE, Piscstaway , NJ, 2018), pp. 4510–4520.

[9] P. Pant and A. B. Farimani, Deep learning for efficient reconstruction of high-resolution turbulent DNS data, arXiv:2010.11348 (2020).

[10] B. Liu, J. Tang, H. Huang, and X.-Y. Lu, Deep learning methods for super-resolution reconstruction of turbulent flows, Phys. Fluids **32**, 025105 (2020).

[11] Z. Zhou, B. Li, X. Yang, and Z. Yang, A robust super-resolution reconstruction model of turbulent flow data based on deep learning, Comput. Fluids **239**, 105382 (2022).

[12] M. Hassanaly, A. Glaws, K. Stengel, and R. N. King, Adversarial sampling of unknown and high-dimensional conditional distributions, J. Comput. Phys. **450**, 110853 (2022).

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks, Commun. ACM **63**, 139 (2020).

[14] Z. Deng, C. He, Y. Liu, and K. C. Kim, Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework, Phys. Fluids **31**, 125111 (2019).

[15] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA* (IEEE, Piscstaway , NJ, 2018), pp. 105–114.

[16] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, ESRGAN: Enhanced super-resolution generative adversarial networks, *Computer Vision – ECCV 2018 Workshops*, edited by L. Leal-Taixé, and S. Roth (Springer International Publishing, Cham, 2019), pp. 63–79.

[17] S. Lee and D. You, Data-driven prediction of unsteady flow over a circular cylinder using deep learning, J. Fluid Mech. **879**, 217 (2019).

[18] Q. Xu, Z. Zhuang, Y. Pan, and B. Wen, Super-resolution reconstruction of turbulent flows with a transformer-based deep learning framework, Phys. Fluids **35**, 055130 (2023).

[19] M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, and H. Pitsch, Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows, Proc. Combust. Inst. **38**, 2617 (2021).

[20] M. Bode, M. Gauding, D. Goeb, T. Falkenstein, and H. Pitsch, Applying physics-informed enhanced super-resolution generative adversarial networks to turbulent premixed combustion and engine-like flame kernel direct numerical simulation data, Proc. Combust. Inst. **39**, 5289 (2023).

[21] L. Nista, C. K. D. Schumann, T. Grenga, A. Attili, and H. Pitsch, Investigation of the generalization capability of a generative adversarial network for large eddy simulation of turbulent premixed reacting flows, Proc. Combust. Inst. **39**, 5279 (2023).

[22] L. Nista, C. K. D. Schumann, T. Grenga, A. N. Karimi, G. Scialabba, M. Bode, A. Attili, and H. Pitsch, Turbulent mixing predictive model with physics-based generative adversarial network, in *10th European Combustion Meeting, Naples, Italy* (RWTH, 2021), pp. 460–465.

[23] T. Grenga, L. Nista, C. K. D. Schumann, A. Karimi, G. Scialabba, A. Attili, and H. Pitsch, Predictive data-driven model based on generative adversarial network for premixed turbulence-combustion regimes, Combust. Sci. Technol. **195**, 3923 (2023).

[24] J. F. MacArt, T. Grenga, and M. E. Mueller, Effects of combustion heat release on velocity and scalar statistics in turbulent premixed jet flames at low and high Karlovitz numbers, Combust. Flame **191**, 468 (2018).

[25] J. F. MacArt, T. Grenga, and M. E. Mueller, Evolution of flame-conditioned velocity statistics in turbulent premixed jet flames at low and high Karlovitz numbers, Proc. Combust. Inst. **37**, 2503 (2019).

[26] K. Zhang, W. Zuo, S. Gu, and L. Zhang, Learning deep CNN denoiser prior for image restoration, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Piscataway, NJ, 2017), pp. 3929–3938.

[27] A. Subramaniam, M. L. Wong, R. D. Borker, S. Nimmagadda, and S. K. Lele, Turbulence enrichment using physics-informed generative adversarial networks, arXiv:2003.01907 (2020).

[28] L. Nista, C. K. D. Schumann, G. Scialabba, T. Grenga, A. Attili, and H. Pitsch, The influence of adversarial training on turbulence closure modeling, in *AIAA SCITECH 2022 Forum* (AIAA, Reston, VA, 2022), pp. 1–9.

[29] L. Mescheder, A. Geiger, and S. Nowozin, Which training methods for GANs do actually converge? *International Conference on Machine Learning* (PMLR, 2018), pp. 3481–3490.

[30] O. Desjardins, G. Blanquart, G. Balarac, and H. Pitsch, High order conservative finite difference scheme for variable density low Mach number turbulent flows, J. Comput. Phys. **227**, 7125 (2008).

[31] T. S. Lundgren, Linearly forced isotropic turbulence, Center for Turbulence Research Annual Research Briefs 2003 (2003), https://web.stanford.edu/group/ctr/ResBriefs03/lundgren.pdf.

[32] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. (O'Reilly Media, Beijing , 2019).

[33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).

[34] A. Sergeev and M. Del Balso, Horovod: Fast and easy distributed deep learning in TensorFlow, arXiv:1802.05799 (2018).

[35] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, On empirical comparisons of optimizers for deep learning, arXiv:1910.05446 (2019).

[36] G. Freytag, J. V. Lima, P. Rech, and P. O. Navaux, Impact of reduced and mixed-precision on the efficiency of a multi-GPU platform on CFD applications, in *Computational Science and Its Applications–ICCSA 2022 Workshops: Malaga, Spain, July 4–7, 2022, Proceedings, Part IV* (Springer, Malaga, Spain, 2022), pp. 570–587.

[37] T. Hrycej, B. Bermeitinger, and S. Handschuh, Training neural networks in single vs double precision, in *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management* (SCITEPRESS–Science and Technology Publications, Valletta, Malta, 2022), pp. 307–314.

[38] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, A dynamic subgrid-scale eddy viscosity model, Phys. Fluids **3**, 1760 (1991).

[39] D. K. Lilly, A proposed modification of the Germano subgrid-scale closure method, Phys. Fluids **4**, 633 (1992).

[40] M. Bode, AI super-resolution subfilter modeling for multi-physics flows, in *Proceedings of the Platform for Advanced Scientific Computing Conference* (ACM, 2023), pp. 1–10.

[41] M. Bode, AI super-resolution: Application to turbulence and combustion, *Machine Learning and Its Application to Reacting Flows: ML and Combustion*, edited by N. Swaminathan, and A. Parente (Springer International Publishing, Cham, 2023), pp. 279–305.

[42] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Piscataway, NJ, 2016), pp. 770–778.

[43] K. Li, S. Yang, R. Dong, X. Wang, and J. Huang, Survey of single image super-resolution reconstruction, IET Image Proc. **14**, 2273 (2020).

[44] A. Jolicoeur-Martineau, The relativistic discriminator: A key element missing from standard GAN, arXiv:1807.00734 (2018).

[45] A. Attili and F. Bisetti, Fluctuations of a passive scalar in a turbulent mixing layer, Phys. Rev. E **88**, 033013 (2013).

[46] S. B. Pope, *Turbulent Flows* (Cambridge University Press, Cambridge, 2000).

[47] R. D. Moser, S. W. Haering, and G. R. Yalla, Statistical properties of subgrid-scale turbulence models, Annu. Rev. Fluid Mech. **53**, 255 (2021).

[48] T. Tirer and R. Giryes, Super-resolution via image-adapted denoising CNNs: Incorporating external and internal learning, IEEE Signal Process. Lett. **26**, 1080 (2019).

[49] K. Duraisamy, Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence, Phys. Rev. Fluids **6**, 050504 (2021).

[50] S. Stolz and N. A. Adams, An approximate deconvolution procedure for large-eddy simulation, Phys. Fluids **11**, 1699 (1999).

[51] Q. Zhao, G. Jin, and Z. Zhou, Deep learning method for the super-resolution reconstruction of small-scale motions in large-eddy simulation, AIP Adv. **12**, 125304 (2022).

[52] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (IEEE, Piscataway, NJ, 2017), pp. 136–144.

[53] L. Xu, J. S. Ren, C. Liu, and J. Jia, Deep convolutional neural network for image deconvolution, in *Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (Curran Associates, 2014), Vol. 27, pp. 1–13.