




RESEARCH ARTICLE | DECEMBER 04 2025

Multiscale super-resolution reconstruction of fluid flows with deep neural networks

G. C. Yang ; R. Y. Luo; Q. H. Yao ; P. J. Wang; J. X. Zhang 

AIP Advances 15, 125209 (2025)

<https://doi.org/10.1063/5.0303157>

Articles You May Be Interested In

A deep learning super-resolution model for turbulent image upscaling and its application to shock wave–boundary layer interaction

Physics of Fluids (February 2024)

Ultra-scaled deep learning temperature reconstruction in turbulent airflow ventilation

Physics of Fluids (June 2024)

Advancing super-resolution of turbulent velocity fields: An artificial intelligence approach

Physics of Fluids (March 2025)

Multiscale super-resolution reconstruction of fluid flows with deep neural networks

Cite as: AIP Advances 15, 125209 (2025); doi: 10.1063/5.0303157

Submitted: 19 September 2025 • Accepted: 13 November 2025 •

Published Online: 4 December 2025



G. C. Yang,^{1,a)} R. Y. Luo,¹ Q. H. Yao,¹ P. J. Wang,² and J. X. Zhang¹

AFFILIATIONS

¹School of Aeronautics and Astronautics, Sun Yat-sen University, Guangzhou 510275, China

²Institute of Intelligent Ocean Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China

^{a)}Author to whom correspondence should be addressed: yanggch8@mail.sysu.edu.cn

ABSTRACT

We present a novel multiscale super-resolution framework (SRLBM) that applies deep learning directly to the mesoscopic density distribution functions of the lattice Boltzmann method for high-fidelity flow reconstruction. Two neural network architectures, a standard convolutional neural network (CNN) and a deeper residual dense network (RDN), are trained to upscale distribution functions from coarse grids by factors of 2, 4, and 8, and then recover velocity, pressure, and vorticity from a single model. For flow past a single cylinder at $Re = 100$, RDN reduces the mean relative error in distribution functions by an order of magnitude compared to CNN and avoids spurious pressure oscillations and vorticity smoothing that affect interpolation and simpler networks. To examine the generalization ability, both models are trained using data from the flow past two cylinders of diameter d at a spanwise distance between the centers of $1.5d$ and a Reynolds number of 200. They are then applied without retraining to wake configurations with distances ranging from $2.0d$ to $3.0d$. In these tests, the mean errors remain essentially unchanged across all distances. However, RDN consistently produces sharper shear-layer roll-ups and secondary eddies. These results demonstrate that super-resolving mesoscopic distribution functions yield richer and more transferable features than operating on macroscopic fields alone. By integrating kinetic theory with deep learning, SRLBM offers a compelling alternative for fluid flow reconstruction, enabling a single model to simultaneously recover multiple high-fidelity flow fields while substantially reducing computational cost.

© 2025 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0303157>

I. INTRODUCTION

High-resolution flow fields are indispensable for applications ranging from aerodynamic optimization of aircraft, automobiles, and wind turbines to prediction of fluid transport in human airways and blood vessels. Achieving such a resolution in computational fluid dynamics (CFD) requires extremely fine spatial meshes and small time steps to capture the smallest flow scales, yet this requirement becomes computationally prohibitive at high Reynolds numbers.¹ At the same time, preliminary design studies and closed-loop feedback control systems frequently require rapid CFD predictions with acceptable accuracy.² Consequently, when computational resources are limited, reconciling high-fidelity simulation with practical efficiency remains a fundamental challenge in fluid flow modeling.

To alleviate the prohibitive computational and memory demands of direct numerical simulation (DNS), researchers have

developed more efficient paradigms, including large-eddy simulation (LES), Reynolds-averaged Navier–Stokes (RANS) modeling, and reduced-order modeling (ROM). LES reduces cost by applying a spatial filter that removes motions below a prescribed cutoff and directly resolving only the large, energy-containing eddies on coarser grids and larger time steps. The filtered subgrid-scale stresses are then modeled by turbulence closures.^{3,4} RANS achieves even greater efficiency by replacing instantaneous equations with time- or ensemble-averaged forms, thus solving for steady mean fields while modeling all turbulent fluctuations through closure relations.^{5,6} As a result, LES typically delivers fidelity higher than that of RANS, since it computes the dominant eddies responsible for most of the momentum and scalar transport. ROM attains the greatest speed by projecting the flow dynamics onto a low-dimensional basis of coherent structures, often obtained by proper orthogonal decomposition or dynamic mode decomposition, exploiting the fact that many complex flows are governed by a few dominant modes.^{7–9}

However, ROMs often depend on bases tailored to specific flow configurations, which constrains their robustness and generality across different geometries and operating conditions.¹⁰

In recent years, machine learning has emerged as a transformative paradigm for enhancing both the accuracy and efficiency of traditional CFD methods. Stevens and Colonius¹¹ trained a neural network to correct truncation errors in a fifth-order shock-capturing scheme, markedly sharpening its resolution of discontinuities. Ajuria Illarramendi *et al.*¹² then introduced a hybrid CFD-convolutional neural network (CNN) framework that learns an approximate pressure Poisson solver, thereby reducing the iterative cost of incompressible flow simulations. Jiang *et al.*¹³ built on this idea by combining a residual neural network with an iterative correction loop, achieving an order-of-magnitude speedup in solving Poisson's equation without compromising accuracy. In the Reynolds-averaged Navier–Stokes context, Ling *et al.*¹⁴ embedded Galilean invariance into a neural network architecture to predict the full anisotropic Reynolds stress tensor, outperforming classical linear and nonlinear eddy viscosity models. Data-driven subgrid scale closures for LES have been realized by training deep networks to map coarse-resolved flow fields to closure terms.^{15–17} In reduced order modeling, autoencoder networks compress high-dimensional flow fields into low-dimensional latent spaces, where subsequent dynamics are predicted via purely data-driven or hybrid physics-informed models, yielding substantial computational savings.^{18–22} Similar techniques have recently been applied to the lattice Boltzmann method, where learned compression schemes and data-driven closure corrections promote both speedups and fidelity enhancements.^{23–25} Collectively, these advances illustrate how machine learning can integrate data-driven insights with governing equations to bridge the fidelity-efficiency gap in fluid-flow simulation.

Machine learning based flow reconstruction has emerged as a powerful alternative for recovering high-fidelity fields from sparse or coarse data. Inspired by breakthroughs in computer vision, Fukami *et al.*^{26,27} pioneered the application of neural networks to super-resolution (SR) turbulent flow fields, demonstrating that learned SR models outmatch bicubic interpolation by accurately recovering high-wavenumber structures without excessive smoothing. Subsequently, Liu *et al.*²⁸ introduced a multi-temporal-path CNN that leverages consecutive snapshots to better capture the unsteady eddies of turbulence, and Xu *et al.*²⁹ showed that transformer architectures can simultaneously learn isotropic and anisotropic features in complex geometries. To circumvent the need for high-resolution (HR) ground truth, Gao *et al.*³⁰ embedded conservation laws and boundary conditions directly into a CNN loss function, yielding a physics-informed SR approach that enforces fluid dynamics constraints *a priori*. More recently, Chu *et al.*³¹ leveraged physics-informed neural networks (PINNs) to reconstruct not only velocity but also hidden fields such as pressure from sparse experimental or simulated measurements, an advance that unlocks detailed fluid–structure interaction analyses for applications ranging from underwater vehicles to aerospace systems.

To date, most super-resolution efforts have targeted individual macroscopic fields or treated quantities such as pressure and vorticity separately. Our approach departs from this paradigm by operating on the lattice Boltzmann mesoscopic density distribution functions instead of the derived macroscopic variables. Because each

lattice node stores multiple direction-resolved particle populations, the mesoscopic distributions carry richer, more localized signatures of momentum transfer, shear, and non-equilibrium effects than low-dimensional macroscopic fields. This multi-channel representation increases the degrees of freedom available to a learning model, making subgrid correlations and directional fluxes easier to infer and reconstruct. In addition, once the high-resolution distribution functions are restored, velocity, pressure, and vorticity follow directly from moment relations and spatial derivatives without additional training. This single-model, moment-consistent approach both reduces the need for multiple specialized networks and preserves the intrinsic coupling among flow variables, which streamlines computation and is particularly advantageous for multiphysics problems that require simultaneous, self-consistent access to diverse flow fields.

The remainder of this paper is organized as follows: In Sec. II, we introduce the proposed super-resolution lattice Boltzmann method (SRLBM) and detail the deep neural network architectures used to upscale mesoscopic distribution functions across multiple scales. Section III describes the high-fidelity dataset associated with a flow past a cylinder, outlines the data pre-processing and augmentation procedures, and presents the training and validation results. In Sec. IV, we apply SRLBM to reconstruct macroscopic flow fields, including velocity, pressure, and vorticity, over various upscaling factors and assess its generalization ability to more complex two-cylinder configurations. Finally, Sec. V summarizes our key findings and discusses future directions.

II. SUPER-RESOLUTION LATTICE BOLTZMANN METHOD (SRLBM)

Incompressible flows, most often described by macroscopic fields of velocity and pressure, arise from the collective motion and density fluctuations of countless molecules, i.e., entities whose individual paths are impossible to track. Kinetic theory provides a bridge between these scales by defining mesoscopic distribution functions, which are statistical averages of molecular velocities that retain key information about momentum transfer and density variation. By operating at this intermediate mesoscopic level, one can capture intricate flow phenomena, such as eddy formation and wave propagation, without resorting to full molecular dynamics (see Fig. 1). This multiscale perspective underpins the super-resolution lattice Boltzmann method proposed here.

A. Lattice Boltzmann method

The lattice Boltzmann method describes fluid dynamics through a discrete set of mesoscopic distribution functions $f_i(\mathbf{x}, t)$. Each f_i represents the mass density of particles at position \mathbf{x} and time t traveling with one of the prescribed discrete velocities \mathbf{c}_i . Under the Bhatnagar–Gross–Krook (BGK) approximation,³² collisions relax each f_i toward its local equilibrium f_i^{eq} over a single relaxation time τ . The resulting lattice Boltzmann equation combines collision and streaming in a single update,

$$f_i(\mathbf{x} + \mathbf{c}_i \delta t, t + \delta t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau} [f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)], \quad (1)$$

where δt is the time step. The equilibrium distribution function f_i^{eq} depends on the local macroscopic fluid density ρ and velocity \mathbf{u} ,³³

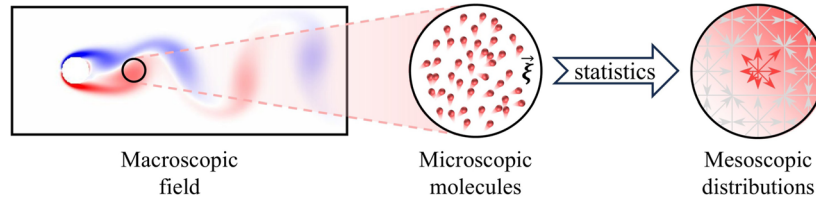


FIG. 1. Multi-scale description of fluid flows. The macroscopic flow field is essentially a collective behavior of microscopic molecules, which can also be represented by a set of mesoscopic density distribution functions through statistical averaging.

$$f_i^{eq}(\rho, \mathbf{u}) = w_i \rho \left[1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2} \right]. \quad (2)$$

Following Qian *et al.*,³³ the D2Q9 lattice structure is adopted in this study for two-dimensional fluid flow simulations. As a result, the speed of sound in lattice units is $1/\sqrt{3}$, and the weighting coefficients w_i become $w_0 = 4/9$ associated with the discrete velocity at rest, $w_i = 1/9$ for $i = 1-4$ streaming in the orthogonal directions, and $w_i = 1/36$ for $i = 5-8$ streaming in the diagonal directions. Macroscopic flow quantities, including density and velocity, can be recovered by taking the first and second velocity moments of the density distribution functions according to the conservation of mass and momentum,

$$\rho(\mathbf{x}, t) = \sum_{i=0}^8 f_i(\mathbf{x}, t), \quad (3)$$

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho} \sum_{i=0}^8 f_i(\mathbf{x}, t) \mathbf{c}_i. \quad (4)$$

Through a multiscale Chapman–Enskog expansion, the Navier–Stokes equations can be derived from the lattice Boltzmann equation, i.e., Eq. (1), resulting in a relationship between the fluid kinematic viscosity ν and the relaxation time τ ,³⁴

$$\nu = c_s^2 \left(\tau - \frac{1}{2} \right) \frac{\delta_x^2}{\delta_t}, \quad (5)$$

where δ_x is the size of a lattice cell. A weak fluid compressibility is allowed in the single relaxation time LBM, and the macroscopic fluid pressure can be obtained from the equation of state $p = c_s^2 \rho$.

The particulate nature of LBM is considered to be advantageous for simulating fluid–structure interactions. In this study, the partially saturated cell (PSC) method originally proposed by Noble and Torczynski³⁵ has been adopted to treat solid obstacles inside a flow. The PSC method generalizes the lattice Boltzmann equation by introducing a new collision operator that is capable of describing the fluid and solid behaviors in a unified manner,

$$f_i(\mathbf{x} + \mathbf{c}_i \delta_t, t + \delta_t) - f_i(\mathbf{x}, t) = [1 - B(\phi, \tau)] \Omega_i^f + B(\phi, \tau) \Omega_i^s, \quad (6)$$

where $B(\phi, \tau)$ is a factor that weighs the relative importance of fluid and solid dynamics of a lattice cell, which depends on the solid volume fraction ϕ and the relaxation time τ ,

$$B(\phi, \tau) = \frac{\phi(\tau - 1/2)}{(1 - \phi) + (\tau - 1/2)}. \quad (7)$$

The weighting factor B increases from 0 to 1 as the solid volume fraction ϕ increases from 0 to 1. The fluid collision operator Ω_i^f takes the form of BGK, i.e., the right hand side of Eq. (1). The solid collision operator, Ω_i^s , is defined with the bounce back of the non-equilibrium density distribution functions to ensure no-slip between fluid and solid,

$$\Omega_i^s = f_{-i}(\mathbf{x}, t) - f_{-i}^{eq}(\rho, \mathbf{u}) + f_i^{eq}(\rho, \mathbf{u}_s) - f_i(\mathbf{x}, t), \quad (8)$$

where \mathbf{u}_s denotes the solid velocity at the position of the lattice cell \mathbf{x} , which is set to zero since, in this study, no moving solid is considered. The subscript $-i$ represents the opposite direction of i .

The PSC method preserves the locality of the nonlinear collision operator while retaining the linearity of the nonlocal streaming process. By allowing the weighting factor B to continuously vary between 0 and 1, partially saturated cells produce a more accurate fluid–solid interface than the staircase boundary that results from a traditional bounce-back scheme, which usually places the interface midway between neighboring lattice nodes.

B. Multiscale super-resolution neural networks

We use supervised learning to train our neural networks for flow super-resolution, which requires sufficient paired low-resolution inputs and high-resolution outputs. In the lattice Boltzmann method, each fluid cell recovers its macroscopic properties from mesoscopic density distribution functions. Consequently, rather than directly super-resolving macroscopic fields such as velocity or vorticity, as in previous studies,^{26,28} we enhance the resolution of density distribution functions (see Fig. 2) by

$$f_i^{SR} = \mathcal{F}(f_i^{LR}, \theta), \quad (9)$$

where f_i^{LR} and f_i^{SR} represent the low-resolution input and the high-resolution output, respectively. All learnable parameters of a neural network are denoted by θ . The neural network works as a nonlinear mapping function \mathcal{F} . Then, the training process becomes essentially an optimization task such that

$$\theta = \arg \min_{\theta} \mathcal{L}(\mathcal{F}(f_i^{LR}, \theta), f_i^{HR}), \quad (10)$$

where f_i^{HR} represents the high-resolution reference data and \mathcal{L} is the loss function. Once the training process completes and the loss converges to a sufficiently small value, the optimized θ can be substituted into Eq. (9) for fast super-resolution reconstructions of

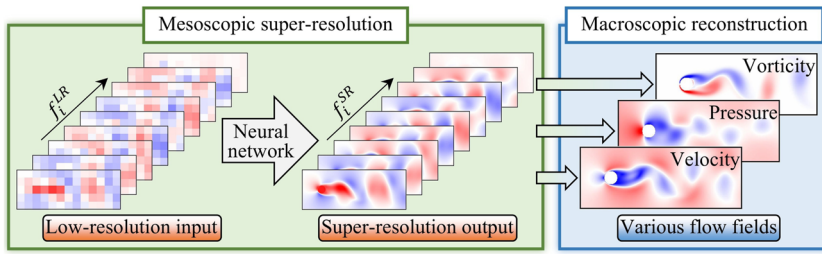


FIG. 2. Multiscale super-resolution reconstruction of macroscopic fluid flows based on the mesoscopic density distribution functions from lattice Boltzmann simulations.

density distribution functions. After that, the super-resolution versions of the macroscopic density (ρ^{SR}) and velocity (\mathbf{u}^{SR}) fields can be obtained,

$$\rho^{SR} = \sum_{i=0}^8 f_i^{SR}, \quad (11)$$

$$\mathbf{u}^{SR} = \frac{1}{\rho} \sum_{i=0}^8 f_i^{SR} \mathbf{c}_i. \quad (12)$$

In addition, ρ^{SR} can be further applied to reconstruct the super-resolution pressure (p^{SR}) field according to the equation of state. The super-resolution vorticity field can be calculated from the corresponding velocity field by using a second-order finite difference scheme.

We denote this multiscale super-resolution flow reconstruction framework as the super-resolution lattice Boltzmann method (SRLBM). In SRLBM, the state of each lattice cell is represented by a vector of mesoscopic density distribution functions, an arrangement directly analogous to a multi-channel image pixel, which combines multiple color channels (e.g., red, green, and blue) into a single visual value.²⁴ This formal equivalence allows many existing image super-resolution architectures to be adapted for SRLBM with only minimal changes to their input and output channel dimensions.

Note that the current SRLBM framework is limited to super-resolution flow reconstruction from coarse inputs, and coupling reconstructed distributions into LBM time stepping is out of scope.

In this work, we implement and compare two such architectures, a standard convolutional neural network (CNN) and a deeper residual dense network (RDN), to evaluate their relative performance in reconstructing high-fidelity flow fields across scales. Furthermore, in the present framework, boundary conditions are implicitly embedded in data rather than by modifying the inner structure of neural networks. Therefore, SRLBM can be trained on data with arbitrary boundary setups, and it generalizes to any combination of boundary conditions, e.g., Dirichlet, Neumann, mixed, and moving walls, without architectural changes. In contrast, super-resolution methods that rely on frequency domain decompositions inherently assume periodicity in each coordinate direction to satisfy basis function orthogonality and, therefore, cannot be directly applied to non-periodic or mixed-type boundary conditions without additional problem-specific treatment.³⁶

1. Convolutional neural network (CNN)

Figure 3(a) depicts the CNN architecture we employ, which consists of two sequential convolutional layers. To generate the low-resolution (LR) input, we first apply bicubic interpolation (BI) to

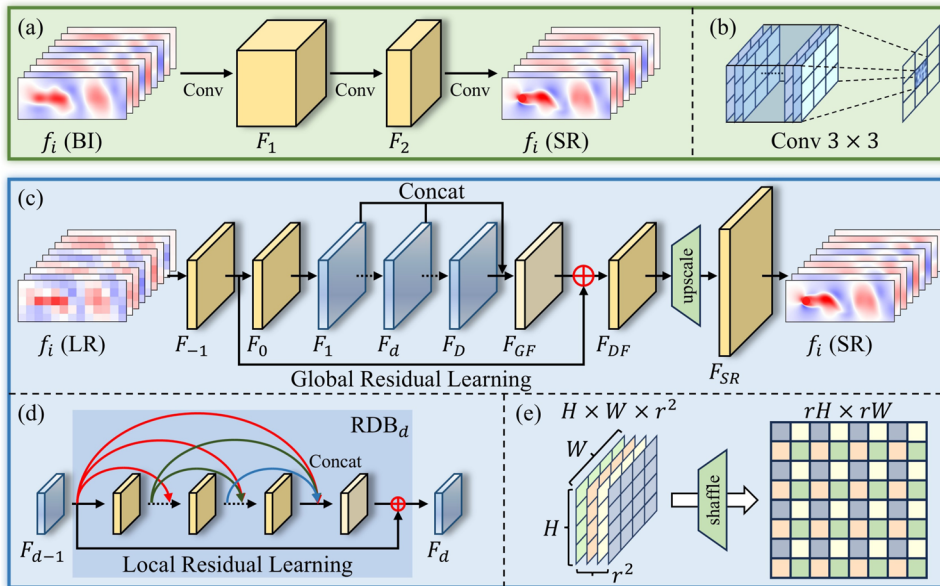


FIG. 3. (a) Architecture for super-resolution reconstruction of density distribution functions based on the convolutional neural network. (b) Schematic of convolution (Conv) operation with a kernel size of 3. (c) Architecture for super-resolution reconstruction of density distribution functions based on the residual dense network. (d) Neural network architecture of a residual dense block. (e) Schematic of the PixelShuffle operation that transforms a tensor of the shape $H \times W \times r^2$ into another tensor of the shape $rH \times rW$.

the high-resolution (HR) density distribution functions, intentionally degrading fine-scale details while preserving the original spatial dimensions. Zero padding is used at each convolutional stage to maintain this fixed size throughout the network.

Under the D2Q9 lattice model (see Sec. II A), the state of each lattice cell comprises nine distribution functions, one for each discrete velocity direction, so the native data naturally form a nine-channel tensor. To provide the network with crucial geometric information, we add a tenth channel encoding the local solid volume fraction, thus distinguishing fluid from solid regions. Thus, the CNN accepts a ten-channel LR tensor and produces nine super-resolved distribution channels, in contrast to conventional image super-resolution networks that typically handle one (grayscale) or three (RGB) channels. Note that the input channel for the solid volume fraction is not shown in Fig. 3 for simplicity.

Following Dong *et al.*,³⁷ our CNN begins with a first convolutional layer (Conv1) that applies a set of learnable filters to the low-resolution input, extracting a feature tensor F_1 that encodes local flow patterns. A second convolutional layer (Conv2) then nonlinearly maps F_1 into high-resolution patch representations F_2 . Figure 3(b) illustrates a single 3×3 convolution operation, and the rectified linear unit (ReLU) activation follows each convolution to introduce nonlinearity and promote sparse, stable feature learning.³⁸ Table I summarizes the detailed CNN configuration, including the number of input and output channels, kernel sizes, and padding values for each layer.

Despite its success in single-image super-resolution, the basic CNN architecture exhibits two critical shortcomings when applied to flow fields. First, it relies on pre-upsampling the low-resolution input, often via bicubic interpolation, to the target grid, a step that inherently smooths fine-scale details and raises computational cost roughly quadratically.³⁹ Second, its strictly sequential stacking of convolutional layers means that only the deepest feature maps inform the final reconstruction, while the rich, multi-scale representations captured in intermediate layers are discarded. By ignoring this hierarchical information, which is vital for flows that span from small eddies to large coherent structures, the standard CNN may struggle to recover the full spectrum of scales, especially when the flow becomes complex (see Sec. IV).

2. Residual dense network (RDN)

Figure 3(c) presents the RDN architecture proposed by Zhang *et al.*,⁴⁰ which effectively addresses the shortcomings of a plain CNN. The network comprises four components: the shallow feature extraction network (SFENet), residual dense blocks (RDBs), dense

feature fusion (DFF), and the up-sampling network (UPNet). In SFENet, two convolutional layers extract shallow features F_{-1} and F_0 . The first feature map, F_{-1} , serves in global residual learning, while the second map, F_0 , is fed into the RDBs.

Let D be the total number of RDBs. We denote the input to the d th block by F_{d-1} and its output by F_d . Each RDB shares the same architecture, which is a sequence of densely connected convolutional layers; see Fig. 3(d). Within a block, every convolutional layer takes as input the concatenation of F_{d-1} and all feature maps produced by the preceding layers in that block. Each output then passes through a ReLU activation. After these layers, local feature fusion (LFF) applies a 1×1 convolution to merge the original input F_{d-1} with the intermediate feature maps. Finally, a local residual learning shortcut adds F_{d-1} element-wise to the fused features, enhancing the representational capacity of the block, as demonstrated in deep residual learning.⁴¹ The output of the final block is denoted by F_D .

After the RDBs extract local dense features, dense feature fusion (DFF) merges these multi-scale representations. First, the outputs of all RDBs are concatenated and then passed through a 1×1 convolution to compress channels, followed by a 3×3 convolution to blend spatial information. This sequence, referred to as global feature fusion (GFF), produces the global feature map F_{GF} . Finally, global residual learning (GRL) combines F_{GF} with the initial shallow feature map F_{-1} via element-wise addition, yielding the dense feature map F_{DF} .

Finally, we employ a sub-pixel convolutional layer, often called PixelShuffle,⁴² as the up-sampling network (UPNet). This layer rearranges the dense feature map F_{DF} of size $H \times W \times n \times r^2$ into a high-resolution feature tensor F_{SR} of size $(rH) \times (rW) \times n$, where r is the upscaling factor and n is the number of feature channels. Figure 3(e) illustrates how PixelShuffle reshapes and permutes these dimensions. A final convolutional layer then reconstructs the super-resolved density distribution functions. Table II details the layer configurations of RDN. As with the CNN, the RDN takes nine density distribution channels (f_0 through f_8) plus a solid volume fraction channel, i.e., ten inputs in total, and produces nine channels corresponding to the super-resolved distributions.

Unlike the plain CNN, which begins by up-sampling its input, the RDN operates directly on low-resolution data, eliminating costly interpolation and streamlining the entire super-resolution workflow. Its densely connected residual blocks produce a rich hierarchy of local features, from the smallest vortices to the largest flow patterns, and then fuse them seamlessly. By fully exploiting this

TABLE I. Detailed settings for the convolutional neural network. The height and width of the patches randomly sampled from the domain for super-resolution are denoted by H and W , respectively.

Layer	Input channels	Output channels	Kernel size	Padding	Output size
Input	$H \times W \times 10$
Conv1	10	64	9	4	$H \times W \times 64$
Conv2	64	32	5	2	$H \times W \times 32$
Output	32	9	5	2	$H \times W \times 9$

TABLE II. Detailed settings for the residual dense network.

Parameters	Value
Input channels	10
Number of RDBs	16
Number of convolutional layers in RDB	8
Number of features	64
Kernel size of convolutional layers for LFF and GFF	1×1
Kernel size of other convolutional layers	3×3
Output channels	9

multiscale feature space, RDN aligns naturally with the scale-spanning character of complex flows, yielding more faithful and detailed high-fidelity flow reconstructions.

3. Network optimization

We train both CNN and RDN with the adaptive moment estimation (Adam) optimizer,²⁶ adjusting all network parameters to minimize a scalar loss that measures the discrepancy between super-resolved outputs and high-resolution ground truth. Although CNNs traditionally use mean square error (MSE) loss³⁷ and RDNs often employ an L_1 norm,⁴⁰ our experiments showed a negligible difference in flow reconstruction accuracy between the two for the test cases presented in this study. For consistency and stable convergence, we, therefore, adopt the MSE loss throughout. Formally, if \tilde{f}_i^{SR} and \tilde{f}_i^{HR} denote the normalized super-resolved and reference distribution functions at each lattice direction i , the loss becomes

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{k=1}^N \left(\sum_{i=0}^8 \left\| \tilde{f}_i^{SR}(\theta) - \tilde{f}_i^{HR} \right\|^2 \right)_k, \quad (13)$$

where N is the number of training samples and k is the counting index. To minimize the loss function, an initial learning rate of 1×10^{-4} is adopted. During training, the learning rate will be reduced by a factor of 10 if there is no improvement after ten consecutive training epochs.

III. DATA ACQUISITION AND MODEL TRAINING

A. Data generation

To generate the training dataset, we perform high-fidelity lattice Boltzmann simulations of a two-dimensional flow past a circular cylinder; see Fig. 4. We denote the cylinder diameter by d and set the computational domain to a streamwise length of $l_x = 30d$ and $l_y = 8d$ in the cross-stream direction. A uniform inlet velocity U is imposed on the left boundary, the right boundary enforces a zero-gradient outlet condition, and periodic boundaries are applied on the top and bottom to minimize confinement effects. The cylinder sits on the horizontal centerline, $5d$ downstream of the inlet, allowing a fully developed wake to form. We choose a spatial resolution of $d/\delta_x = 32$ and a relaxation time of $\tau = 0.53$ to ensure that the wake dynamics is well resolved. Extensive validation against reference drag coefficients, pressure distributions, recirculation lengths, and flow-separation angles at Reynolds numbers $Re = Ud/\nu = 20$

and 40 confirms excellent agreement with the results of Ref. 43. Note that the steady-state results at $Re = 20$ and 40 are only used for the verification of numerical calculations, which are not presented here.

We train both CNN and RDN on data from a fully resolved LBM simulation at $Re = 100$, where vortex shedding produces rich multiscale structures. Although the complete grid spans 960×256 lattice cells, we extract only the region around the cylinder, resulting in an area of interest (AOI) measuring $13d \times 5d$ (416×160 lattice cells, see Fig. 4), to focus learning on the most dynamically active flow. At $Re = 100$, the flow transitions to an unsteady regime with periodic vortex shedding once fully developed. We monitor the drag coefficient $C_d = F_x/(0.5\rho U^2 d\delta_x)$ and the lift coefficient $C_l = F_y/(0.5\rho U^2 d\delta_x)$, where F_x and F_y are the hydrodynamic forces on the cylinder in the streamwise and spanwise directions, respectively. Until $t = 2$ s, the wake remains symmetric about the horizontal centerline, and C_l is near zero (see the inset vorticity at $t = 1$ s in Fig. 5). Beyond $t \approx 2$ s, the symmetry breaks, C_l oscillates between positive and negative peaks, and C_d rises to its established mean, consistent with previous studies.⁴⁴ We, therefore, select unsteady snapshots from $t = 2$ s to $t = 5$ s as our training set, ensuring that the networks learn to reconstruct both small-scale local vortices and large-scale wake patterns.

To enrich our training set without extra simulations, we have applied systematic data augmentation. From each unsteady snapshot, we extract 20 random 64×64 cell patches within the AOI. Each patch then undergoes random horizontal and vertical flips plus 90° rotations, generating geometric variants that preserve the core flow physics. These transformations expose the networks to shifted, reflected, and reoriented vortex patterns, imposing invariance to translations and symmetries in the flow. As a result, model robustness improves and overfitting diminishes, since the networks learn to recognize essential flow features under diverse geometric configurations.⁴⁵

Low-resolution inputs are generated by bicubic downsampling of the high-resolution distribution functions at upscaling factors $r = 2, 4$, and 8. This degradation mimics the smooth loss of fine-scale detail while preserving overall flow structure, following common image SR practices. For each scale, we assemble paired LR–HR patches and then split them randomly into training (80%) and testing (20%) sets. This partitioning ensures that the networks learn robust multiscale mappings while providing an unbiased evaluation of super-resolution performance.

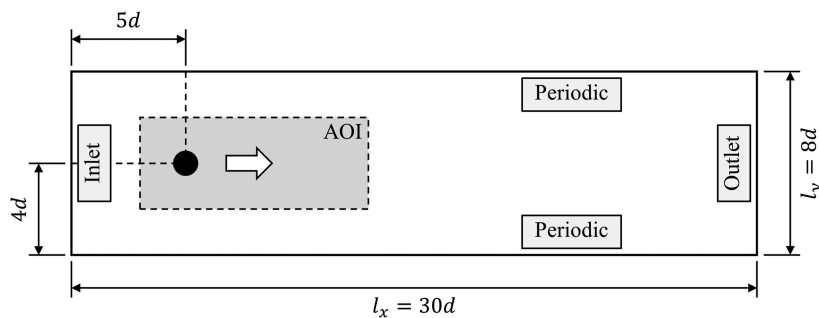


FIG. 4. Sketch of the numerical setup for the lattice Boltzmann simulation of a flow past a fixed cylinder (black circle). The gray area around the cylinder shows the area of interest (AOI), where the flow information is extracted for the training and testing of the machine learning models.

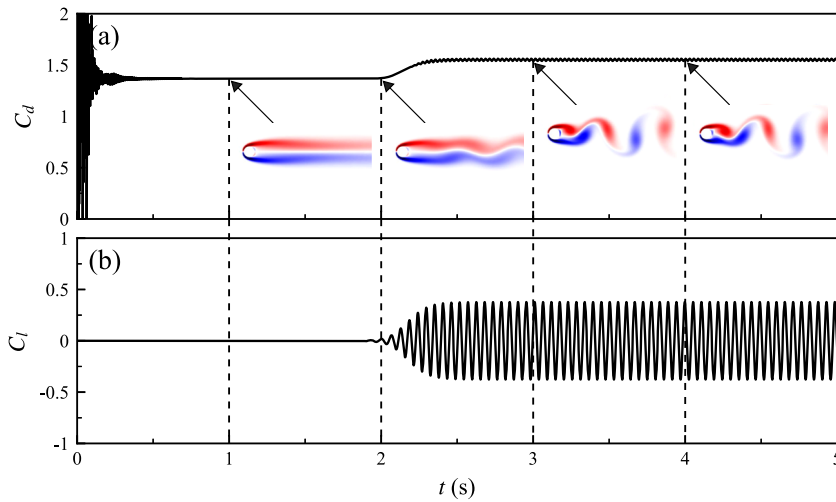


FIG. 5. Results of the lattice Boltzmann simulation of a flow past a fixed cylinder when $Re = 100$. (a) Temporal evolution of the drag coefficient. The insets show the vorticity fields around the cylinder at $t = 1, 2, 3$, and 4 s. (b) Temporal evolution of the lift coefficient.

B. Training results

To ensure sufficient training coverage, we extract varying numbers of snapshots evenly from $t = 2$ – 5 s within the AOI. Figure 6 plots the mean relative error $|(f_i^{SR} - f_i^{HR})/f_i^{HR}|$ of the test set against the number of snapshots for both CNN and RDN. We favor this normalized metric because each distribution f_i scales with its lattice weight w_i [see Eq. (2)], making absolute errors in the dominant rest population f_0 disproportionately large. The shaded bands mark three times the standard deviation in the testing samples. In the low-data regime (fewer than 1000 snapshots), the simpler CNN achieves lower error than the deeper RDN, since its compact architecture requires less data to stabilize. As we increase the dataset, the error of the CNN quickly plateaus, while the error of RDN continues to drop sharply until about 1500 snapshots, after which the gains diminish. At 3000 snapshots, the RDN achieves a mean relative error of

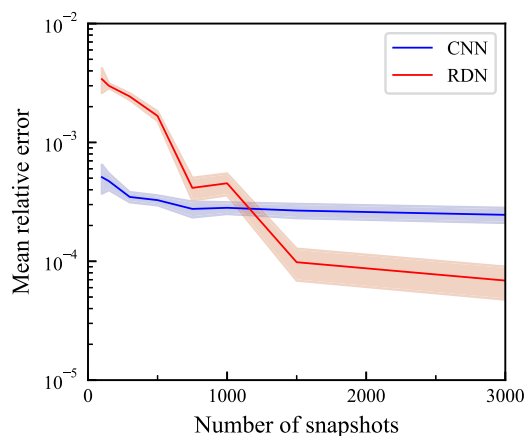


FIG. 6. Plot of the mean relative errors of the CNN and RDN models against the number of snapshots at the super-resolution scale of 8. The error is calculated based on the testing data. The shaded area indicates the range of variations set by three times the standard deviations.

6.88×10^{-5} , significantly below the error of CNN, i.e., 2.46×10^{-4} . Training neural networks with small datasets can be extremely powerful, e.g., by extracting rich flow information from a single turbulent snapshot⁴⁶ or embedding physical priors directly into the learning process,⁴⁷ one can dramatically reduce the data requirements for accurate reconstruction. Furthermore, both CNN and RDN architectures can be flexibly scaled by tuning layer depth, feature channel widths, or the number of residual blocks to find the optimal balance between reconstruction accuracy and computational efficiency under different data budgets. In this study, however, our objective is to benchmark the peak performance of both models, so we train each on a sufficiently large dataset of 3000 snapshots to ensure we evaluate their capabilities at the performance ceiling. Future work can reduce HR data requirements via transfer learning or embedding physical constraints into training while maintaining model performance.

Figure 7 plots the MSE loss trajectories, optimized through Eq. (13), for both CNN and RDN at upscaling factors $r = 2, 4$, and 8 . All curves stabilize by the 400th epoch, indicating convergence of the training process. The final test loss remains consistently smaller than the training loss, which is caused by the heavy data augmentations during training. As the super-resolution scale increases, the asymptotic loss rises, reflecting the growing challenge of reconstructing finer flow details from coarser inputs. Crucially, at every scale, the training and validation losses of RDN remain over an order of magnitude lower than those of the CNN, underscoring the superior capacity of RDN to capture the multiscale features inherent in fluid flows.

Figure 8 compares the nine distribution functions f_0 to f_8 around the cylinder at a randomly selected time instant in the test set for an upscaling factor $r = 8$. For each f_i , we display the high-resolution ground truth, the low-resolution input, its bicubic interpolation, and the super-resolved output from CNN and RDN. All fields are min-max normalized to the range $[0, 1]$ for consistent visualization. Under such extreme downsampling, i.e., only four grid points span the cylinder diameter, the LR fields become heavily pixelated, and BI merely smooths the edges while losing fine details and blurring steep transitions at the fluid–solid interface (third column of Fig. 8).

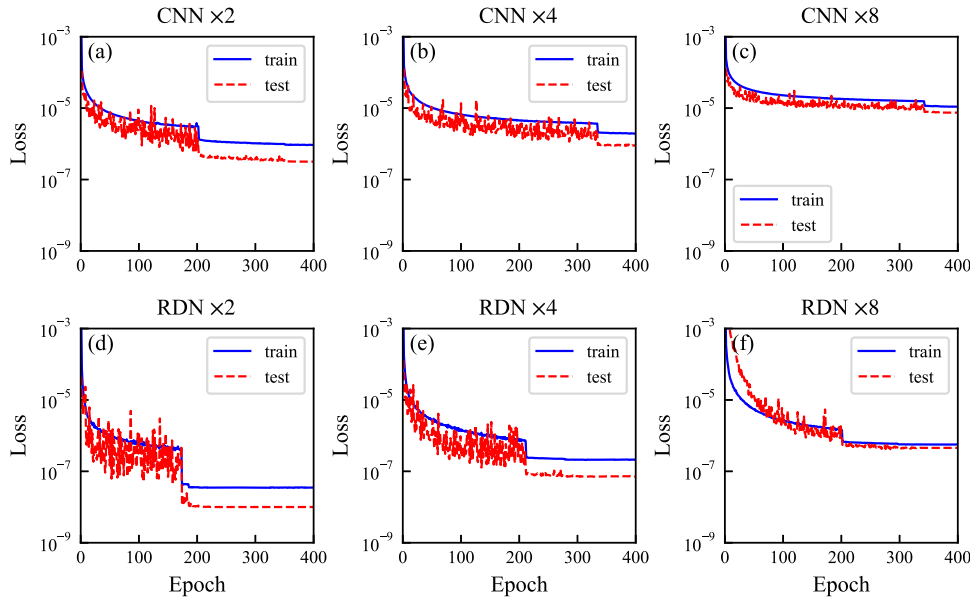


FIG. 7. Learning curves for different models and upscaling factors using 3000 snapshots evenly extracted from $t = 2-5$ s. [(a)–(c)] CNN model with upscaling factors of 2, 4, and 8. [(d)–(f)] RDN model with upscaling factors of 2, 4, and 8.

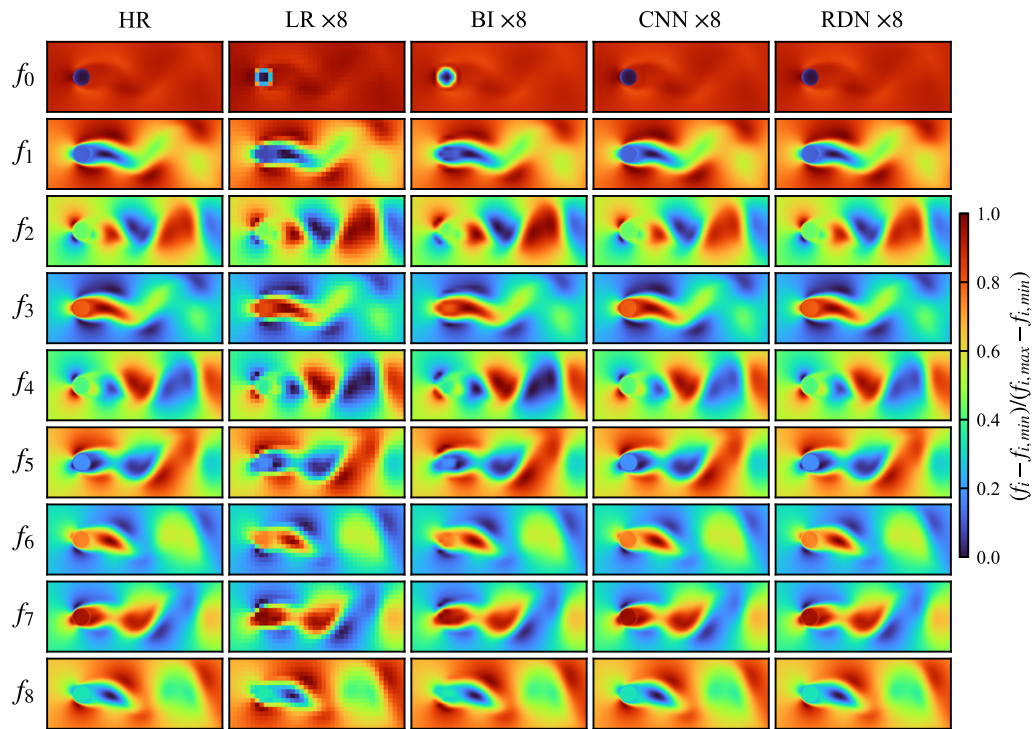


FIG. 8. Comparison of the density distribution functions around the cylinder at a randomly selected time instant in the testing set. The condition with the upscaling factor of 8 is considered. First column: high-resolution (HR) reference data, second column: low-resolution (LR) input data, third column: output data from bicubic interpolation (BI), fourth column: super-resolution output data from CNN, and fifth column: super-resolution output data from RDN.

In contrast, both CNN and RDN reconstructions (fourth and fifth columns of Fig. 8) restore sharp cylinder boundaries and recover coherent wake patterns that closely match the HR reference. This dramatic improvement arises because the networks learn

nonlinear correlations that infer subgrid-scale fluctuations from their surroundings. The dense residual connections of RDN, in particular, enhance subtle gradient transitions and sharpen small-scale features. These results underscore that the learned super-resolution

transcends simple pixel interpolation. Overall, the machine learning models are able to reconstruct multiscale physics encoded in density distribution functions effectively.

IV. MULTISCALE SUPER-RESOLUTION FLOW RECONSTRUCTION

Extending super-resolution to the kinetic realm of density distribution functions f_i , rather than operating on derived macroscopic variables, we obtain two transformative benefits. First, the lattice distributions span a finer granularity, with each cell comprising nine directional populations, so the networks inherit richer flow physics from vortices to subtle shear layers. Embedding this mesoscopic detail into the training lets the models infer sharper, more physically consistent reconstructions. Second, once we predict the full set of f_i , we can derive multiple macroscopic quantities, such as velocity, pressure, and vorticity, without retraining separate networks for each variable. In what follows, we demonstrate this multiscale framework on flow past a single cylinder, highlighting its reconstruction fidelity, and on two-cylinder configurations, probing the ability of the models to generalize across more complex wake interactions.

A. Flow past one cylinder

Figure 9 presents the high-resolution, low-resolution, and super-resolved velocity fields for upscaling factors $r = 2, 4$, and 8. In each panel, the root mean square error (RMSE) between the reconstructed and the HR fields is annotated at the top-right corner if applicable, while a zoomed inset focuses on the steep velocity gradients adjacent to the cylinder. Note that all errors are measured against the high-resolution LBM solutions; lower error directly indicates closer agreement with a physically consistent solver reference and, thus, better satisfaction of the governing equations and conserved moments encoded in the HR data. As r increases, all methods

experience growing errors, reflecting the progressive loss of fine-scale structures in coarser inputs. However, both the CNN and the RDN models outperform bicubic interpolation at every scale. RDN excels in general, achieving the lowest RMSE, likely due to its deeper architecture and dense residual connections that capture multiscale features more effectively; see Sec. II. Crucially, bicubic interpolation blurs the sharp velocity jump at the fluid–solid interface, especially at $r = 8$, while CNN and RDN preserve sharp boundary layers, with RDN delivering the most faithful restoration of wake dynamics.

Pressure, derived from the same set of density distribution functions as velocity, proves even more sensitive to downsampling artifacts. In Fig. 10, we present the high-resolution, low-resolution, and reconstructed pressure fields across upscaling factors $r = 2, 4$, and 8. As r increases, bicubic interpolation incurs the largest errors, while RDN consistently delivers the most accurate reconstructions. At $r = 2$, interpolation produces an unphysical white halo, implying zero pressure around the cylinder, whereas the true field features a positive stagnation peak upstream and negative suction in the wake. At $r = 8$, CNN reconstruction exhibits spurious pressure oscillations absent in both reference data and RDN output. Although the RMSE of RDN (0.111) is slightly smaller than the RMSE of CNN (0.136) at $r = 8$, its pressure contours, especially near the cylinder surface, align far more closely with the ground truth. This contrast underscores that global metrics such as RMSE can obscure critical local discrepancies, which are essential to accurately capture pressure-driven forces and boundary layer dynamics.

To further quantify reconstruction accuracy, we compute the pressure coefficient $C_p = p/(0.5\rho U^2)$ around the cylinder for both the high-resolution reference and the super-resolved fields. Figure 11 plots C_p as a function of the polar angle α (with $\alpha = 0^\circ$ at the stagnation point and $\alpha = 180^\circ$ on the lee side) for the upscaling factors $r = 2, 4$, and 8. Bicubic interpolation is omitted here because of its excessively large deviations.

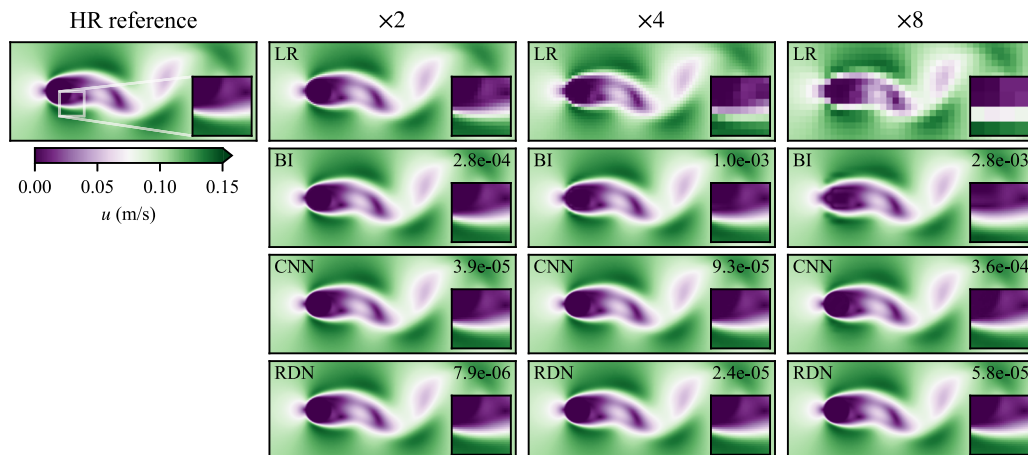


FIG. 9. Velocity field of the flow past a cylinder. First column: high-resolution (HR) velocity field. Second column: low-resolution (LR), bicubic interpolated (BI), and super-resolution (CNN and RDN) velocity fields when the upscaling factor is 2. Third column: LR, BI, CNN, and RDN velocity fields when the upscaling factor is 4. Fourth column: LR, BI, CNN, and RDN velocity fields when the upscaling factor is 8. The number at the top-right corner indicates the root-mean-square error, if applicable. A zoom-in view of a region close to the cylinder is shown as an inset at the bottom-right corner. All subplots share the same colorbar.

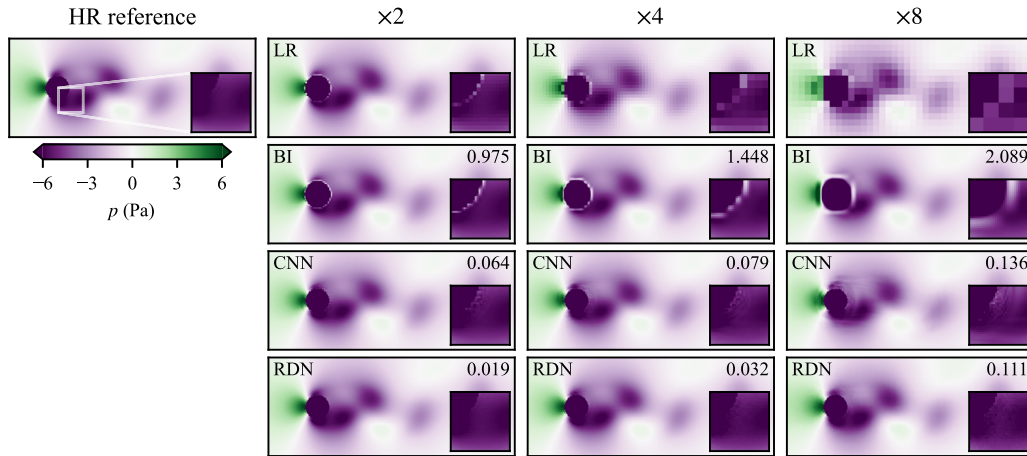


FIG. 10. Pressure field of the flow past a cylinder. The same arrangements of subplots as Fig. 9 are adopted.

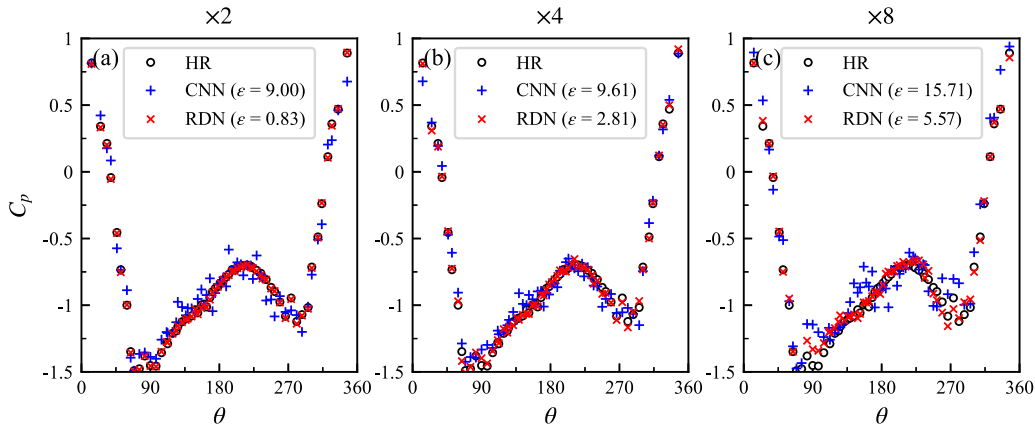


FIG. 11. Comparison of the high-resolution (HR) and super-resolution (CNN and RDN) pressure distributions around the cylinder wall at a randomly selected time instant in the testing dataset when $Re = 100$. (a) Upscaling factor of 2. (b) Upscaling factor of 4. (c) Upscaling factor of 8. The pressure coefficient C_p equals the pressure p normalized by $0.5\rho U^2$.

At $r = 2$, RDN reproduces the entire pressure profile along the fluid–solid interface with remarkable fidelity. Its total absolute error, $\epsilon = 0.83$, is an order of magnitude smaller than that of CNN ($\epsilon = 9.00$). In contrast, the suction peaks of CNN behind the cylinder exhibit spurious oscillations, revealing its limited ability to infer strong negative pressures. When $r = 4$, the error of RDN grows noticeably at angles around $\alpha = 90^\circ$ and 270° , where the true pressure gradient changes rapidly. Meanwhile, the overall error of CNN remains nearly unchanged, reflecting its systematic smoothing of sharp features. At the extreme scale $r = 8$, both networks struggle as the oscillating wake induces complex pressure fluctuations across $\alpha \in (90^\circ, 270^\circ)$, yet RDN still outperforms CNN in capturing the dominant trend. These results underscore the capacity of RDN to reconstruct critical pressure distributions even under severe downsampling, demonstrating the potential of SRLBM for efficient high-fidelity fluid–structure interaction simulations.

Another crucial macroscopic flow quantity is the vorticity, which is calculated based on the derivatives of the velocities, i.e., $\omega = \partial u_y / \partial x - \partial u_x / \partial y$, where u_x and u_y are the flow velocities in the x and y directions, respectively. According to Fukami *et al.*,²⁶ the vorticity field is amplified by the wavenumber compared to the velocity field, which stresses the presence of small scale flow structures. The vorticity fields reconstructed by various methods are compared in Fig. 12. Again, the reconstruction error increases as the upscaling factor increases, and RDN is the most accurate method compared to CNN and bicubic interpolation. Take the results with an upscaling factor of 8 as an example; it is rather encouraging that the machine learning models are capable of restoring the major flow structures close to the cylinder, while the low resolution input appears to lose most of the flow information; see the last column of Fig. 12 and the zoom-in views. Note that the frame is selected from the test set, which is not used for training the machine learning models.

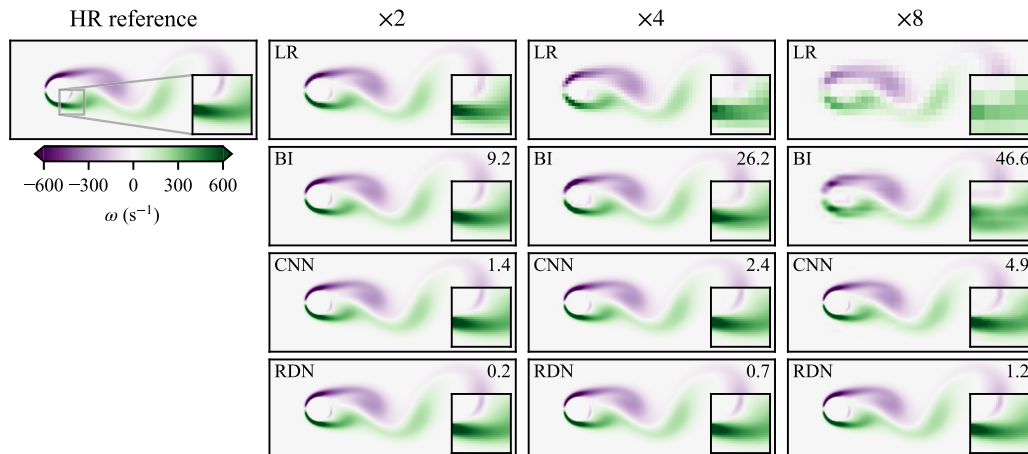


FIG. 12. Vorticity field of the flow past a cylinder. The same arrangements of subplots as Fig. 9 are adopted.

In contrast, the interpolation method cannot recover extreme negative and positive vorticity values around the cylinder, further demonstrating the effectiveness of the machine learning models.

Overall, comparing CNN and RDN, the dense residual blocks and late-stage fusion of RDN preserve and propagate multi-scale, direction-resolved features from the f_i channels, enabling more accurate recovery of high-wavenumber vorticity and stable pressure fields, whereas the pre-upsampling and shallow sequential mapping of CNN tend to smooth or misrepresent small-scale flow features.

B. Flow past two cylinders

Evaluating the ability of machine learning models to generalize beyond their training conditions is crucial for practical super-resolution of fluid flows. In earlier work, we showed that a CNN trained at a single Reynolds number can successfully reconstruct flows at other close Reynolds numbers.⁴³ Here, we extend that test to a fundamentally new wake topology, i.e., two identical cylinders placed side-by-side in the spanwise direction following Morimoto *et al.*⁴⁸ The proximity of the cylinders induces complex wake-wake interactions, ranging from gap flows to synchronized vortex shedding, which depend sensitively on their spacing. By applying both CNN and RDN to this two-body configuration, we probe the capacity of each network to infer flow features it has never encountered during training, thus rigorously assessing their robustness and adaptability.

Figure 13 evaluates both networks on an unseen snapshot of flow past two side-by-side cylinders (the center-to-center distance is $1.5d$) at $Re = 200$, using an upscaling factor of 8. The reconstructed velocity, pressure, and vorticity fields reveal that the wake interference produces markedly more intricate patterns than the single-cylinder case. In the velocity field, RDN achieves an error roughly an order of magnitude below that of CNN, yet CNN still captures the overall shear and recirculation with decent fidelity.

The pressure reconstructions expose lingering deficiencies of CNN, i.e., spurious oscillations around each cylinder surface,

whereas RDN aligns closely with the high-resolution reference, although with slightly elevated noise in the zoomed inset. This suggests that the dense residual pathways of RDN better stabilize steep pressure gradients under complex wake interactions. Vorticity proves to be the most rigorous test. The super-resolved vorticity field from CNN is both blurred and incomplete, with its RMSE exceeding 40. In contrast, RDN delivers fidelity, accurately positioned vorticity lobes with an RMSE of just 6.9, faithfully restoring the alternating eddies that velocity alone cannot reveal.

Efforts to deepen or widen CNN yielded only marginal gains, confirming that its parameter capacity is already fully exploited and that its representational ability has saturated. Only the hierarchical fusion of local and global features of RDN, enabled by dense connections and residual learning, provides the representational power necessary for super-resolving truly complex and multiscale flow fields.

To probe true generalization, we challenged both CNN and RDN, each trained only with the $1.5d$ -distance case, to reconstruct flows at new separations of $2.0d$, $2.5d$, and $3.0d$ between the cylinder centers. Figure 14 shows a representative vorticity field at an extreme upscaling factor of 8. Here, the bicubically downsampled input is so coarse that coherent vortical structures almost vanish. Remarkably, both models infer the correct wake topology, recapturing alternating vorticity cores and shear layers between cylinders, despite never seeing these geometries during training. However, RDN consistently unveils sharper shear-layer roll-ups and secondary eddies, attesting to its hierarchical fusion of local and global features. This contrasts with the smoother and less detailed reconstructions from CNN and underscores the superior ability of RDN to extrapolate complex spacing-dependent flow physics.

Figure 15 shows the mean relative error (MRE) of the density distribution functions as the cylinder distance varies from $1.5d$ to $3.0d$ for upscaling factors $r = 2, 4$, and 8 . Interestingly, both CNN and RDN exhibit MREs that are essentially invariant to cylinder distance, demonstrating that models trained on the most challenging case, tight spacing with strong wake interference, seamlessly

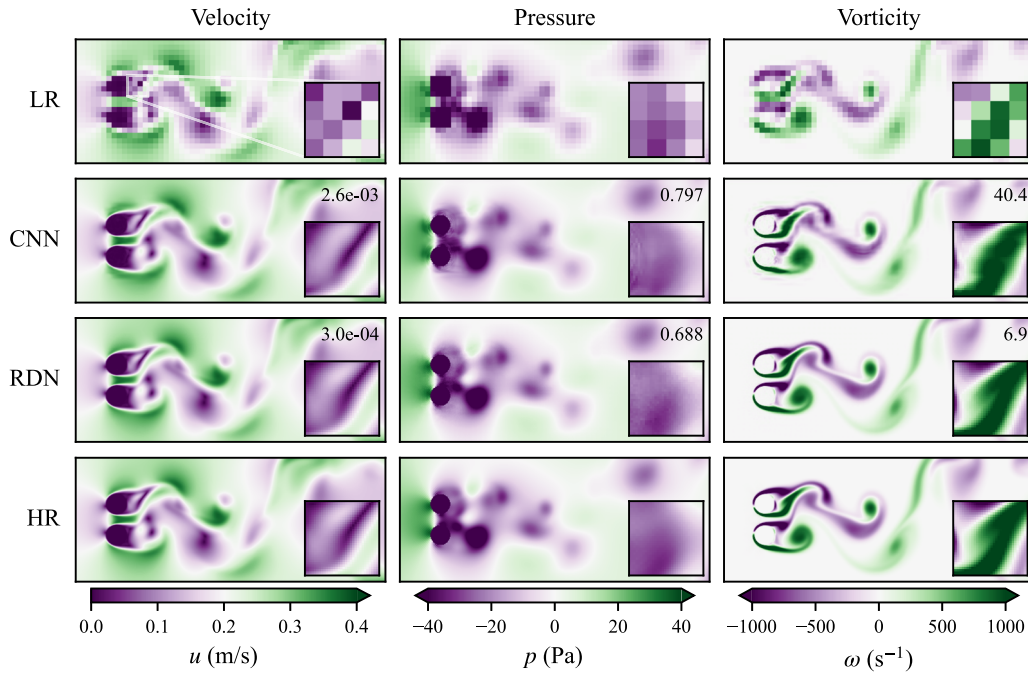


FIG. 13. Low-resolution (LR), super-resolution (CNN and RDN), and high-resolution (HR) flow fields of the flow past two cylinders when $Re = 200$. First column: velocity field, second column: pressure field, and third column: vorticity field. The upscaling factor is 8. The distance between the cylinder centers is $1.5d$. The number at the top-right corner indicates the root-mean-square error, if applicable. A zoom-in view of a region behind the cylinders is shown as an inset at the bottom right corner.

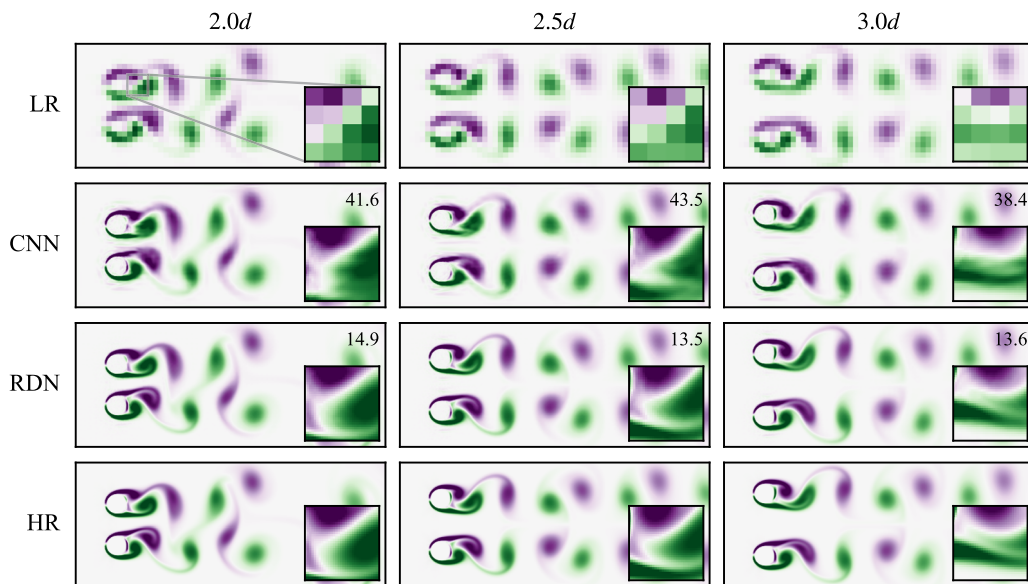


FIG. 14. Low-resolution (LR), super-resolution (CNN and RDN), and high-resolution (HR) vorticity fields of the flow past two cylinders separated by different distances when $Re = 200$. First column: $2.0d$, second column: $2.5d$, and third column: $3.0d$. The upscaling factor is 8. The super-resolution models are trained with the data obtained at $1.5d$. The number at the top-right corner indicates the root-mean-square error, if applicable. A zoom-in view of a region behind the cylinders is shown as an inset at the bottom right corner. The same colorbar for vorticity in Fig. 13 has been adopted.

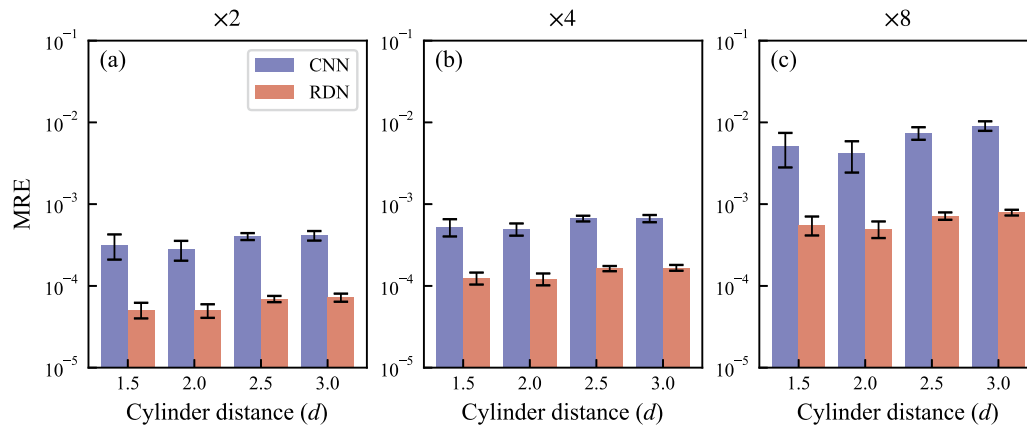


FIG. 15. Comparison of the mean relative errors (MREs) of CNN and RDN regarding the density distribution functions at different upscaling factors: (a) 2, (b) 4, and (c) 8. The machine learning models are trained with the data obtained at $1.5d$. The Reynolds number is fixed at 200. The error bars indicate the standard deviations.

generalize to more benign configurations. This robustness contrasts sharply with approaches that reconstruct macroscopic fields directly, where errors can increase by more than an order of magnitude when geometry changes.⁴⁸ We attribute the stability of SRLBM to three key factors: (i) mesoscopic inputs provide richer, transferable physics across scales; (ii) patch-based training and data augmentation, which expose the models to a wide variety of local flow patterns and cylinder positions (see Sec. III A); and (iii) fully convolutional architectures impart shift-equivariance, so learned features generalize to cylinders located anywhere in the domain. Thus, even without explicit location conditioning, as in many generative model approaches, our supervised CNN and RDN robustly reconstruct wakes around bluff bodies solely by leveraging mesoscopic distribution learning.

The wider error bars at $1.5d$ and $2.0d$ reflect the higher variability inherent in strongly interacting wakes, whereas the flows at $2.5d$ and $3.0d$ are more periodic and predictable, resulting in tighter uncertainty bounds. Across every spacing and scale, RDN achieves MREs roughly an order of magnitude below those of CNN, translating directly into more faithful super-resolution of macroscopic flow quantities as discussed above.

V. CONCLUSIONS

In this work, we have introduced SRLBM, a novel multi-scale super-resolution framework that operates directly on the mesoscopic density distribution functions of the lattice Boltzmann method rather than on derived macroscopic variables. By leveraging the greater degrees of freedom inherent in discrete populations per lattice site, SRLBM encodes richer physics into a single deep network and then recovers velocity, pressure, and vorticity simultaneously. This unified strategy not only streamlines computation but also preserves the natural coupling among flow variables, a feature that is essential for multiphysics simulations requiring concurrent access to diverse flow fields.

Through a systematic comparison of a standard convolutional neural network and a residual dense network, both trained on high-resolution flow past a cylinder data at a Reynolds number of 100,

we have demonstrated that RDN delivers superior fidelity across all upscaling factors. The mean relative error of RDN on the distribution functions is an order of magnitude lower than that of CNN once enough training samples are available. When reconstructing macroscopic quantities at upscaling factors up to 8, RDN avoids spurious pressure oscillations and faithfully restores fine-scale vorticity structures, whereas CNN, despite its adequate performance on velocity, exhibits smoothing and artifacts under severe downsampling.

We have further challenged these models with unseen wake configurations by applying them to flows past two side-by-side cylinders at separations of $2.0d$, $2.5d$, and $3.0d$ without retraining (trained using only $1.5d$ data). Both networks recover the correct wake topology even when faced with very coarse input fields ($r = 8$), yet RDN consistently resolves sharper shear-layer roll-ups and secondary eddies. Moreover, the mean relative error of both models remains essentially invariant to cylinder spacing, in contrast to traditional macroscopic field based methods whose errors escalate by orders of magnitude when geometry changes. This robustness attests to the power of learning based on mesoscopic distributions, which carry transferable multiscale information that facilitates generalization.

Looking forward, the SRLBM paradigm opens several promising avenues, yet this study is limited to low Reynolds number flows, and validating the framework at higher Reynolds regimes will be an important next step. Although for the moderate upscaling factors studied here (2, 4, and 8), a purely data-driven RDN already reconstructs decent flow patterns without explicit physical constraints, embedding conservation laws and explicit boundary constraints into the network could further improve accuracy and reduce training data. State-of-the-art super-resolution innovations in computer vision, such as attention-based transformers²⁹ and generative adversarial networks,⁴⁹ can be adapted to enrich feature representation and sharpen reconstructed flow details. Although our implementation here relies on LBM-derived numerical snapshots, the ability to recover mesoscopic distribution functions from macroscopic flow measurements means SRLBM can also fuse numerical and experimental data, enabling physics-guided convergence across various data sources.⁵⁰ It is also worthwhile to evaluate the model performance across a range of Reynolds numbers, obstacle shapes,

and dimensions in a follow-up study. Overall, this work establishes an LBM-based, multiscale framework that offers a compelling alternative for achieving fast and high-fidelity super-resolution reconstructions for fluid dynamics problems.

ACKNOWLEDGMENTS

This research was developed under the support of the National Key Research and Development Program of China, Grant No. 2020YFA0712500, and the Guangdong Basic and Applied Basic Research Foundation, Grant Nos. 2025A1515012962, 2023A1515012881, and 2022B1515120009. The authors thank D. W. David Wang for his insightful discussions on machine learning models for image super-resolution and Xu Han for his support in reviewing this manuscript prior to submission.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

G. C. Yang: Conceptualization (lead); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (lead); Supervision (lead); Visualization (equal); Writing – original draft (lead). **R. Y. Luo:** Data curation (lead); Investigation (equal); Methodology (equal); Validation (equal); Visualization (equal); Writing – review & editing (equal). **Q. H. Yao:** Funding acquisition (equal); Project administration (equal); Resources (equal); Supervision (equal); Writing – review & editing (equal). **P. J. Wang:** Funding acquisition (equal); Project administration (equal); Resources (equal); Writing – review & editing (equal). **J. X. Zhang:** Funding acquisition (equal); Project administration (equal); Resources (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- H. Choi and P. Moin, “Grid-point requirements for large eddy simulation: Chapman’s estimates revisited,” *Phys. Fluids* **24**, 011702 (2012).
- R. Vinuesa and S. L. Brunton, “Emerging trends in machine learning for computational fluid dynamics,” *Comput. Sci. Eng.* **24**, 33–41 (2022).
- J. Smagorinsky, “General circulation experiments with the primitive equations: I. The basic experiment,” *Mon. Weather Rev.* **91**, 99–164 (1963).
- M. Lesieur and O. Metais, “New trends in large-eddy simulations of turbulence,” *Annu. Rev. Fluid Mech.* **28**, 45–82 (1996).
- H. C. Chen, V. C. Patel, and S. Ju, “Solutions of Reynolds-averaged Navier–Stokes equations for three-dimensional incompressible flows,” *J. Comput. Phys.* **88**, 305–336 (1990).
- G. Alfonsi, “Reynolds-averaged Navier–Stokes equations for turbulence modeling,” *Appl. Mech. Rev.* **62**, 040802 (2009).
- C. W. Rowley and S. T. M. Dawson, “Model reduction for flow analysis and control,” *Annu. Rev. Fluid Mech.* **49**, 387–417 (2017).
- K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, “Modal analysis of fluid flows: An overview,” *AIAA J.* **55**, 4013–4041 (2017).
- K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. M. Dawson, and C.-A. Yeh, “Modal analysis of fluid flows: Applications and outlook,” *AIAA J.* **58**, 998–1022 (2020).
- R. Vinuesa and S. L. Brunton, “Enhancing computational fluid dynamics with machine learning,” *Nat. Comput. Sci.* **2**, 358–366 (2022).
- B. Stevens and T. Colonius, “Enhancement of shock-capturing methods via machine learning,” *Theor. Comput. Fluid Dyn.* **34**, 483–496 (2020).
- E. Ajuria Illarramendi, A. Alguacil, M. Bauerheim, A. Misdariis, B. Cuenot, and E. Benazera, “Towards an hybrid computational strategy based on deep learning for incompressible flows,” in *AIAA AVIATION 2020 FORUM*, 2020.
- Z. Jiang, Z. Wang, Q. Yao, G. Yang, Y. Zhang, and J. Jiang, “A neural network-based Poisson solver for fluid simulation,” *Neural Process. Lett.* **56**, 233 (2024).
- J. Ling, A. Kurzawski, and J. Templeton, “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance,” *J. Fluid Mech.* **807**, 155–166 (2016).
- M. Gamahara and Y. Hattori, “Searching for turbulence models by artificial neural network,” *Phys. Rev. Fluids* **2**, 054604 (2017).
- A. Beck, D. Flad, and C.-D. Munz, “Deep neural networks for data-driven LES closure models,” *J. Comput. Phys.* **398**, 108910 (2019).
- R. Maulik, O. San, A. Rasheed, and P. Vedula, “Subgrid modelling for two-dimensional turbulence using neural networks,” *J. Fluid Mech.* **858**, 122–144 (2019).
- T. Murata, K. Fukami, and K. Fukagata, “Nonlinear mode decomposition with convolutional neural networks for fluid dynamics,” *J. Fluid Mech.* **882**, A13 (2020).
- K. Lee and K. T. Carlberg, “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders,” *J. Comput. Phys.* **404**, 108973 (2020).
- P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, “Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks,” *Proc. R. Soc. A* **474**, 20170844 (2018).
- J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, “Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach,” *Phys. Rev. Lett.* **120**, 024102 (2018).
- J. L. Callahan, S. L. Brunton, and J.-C. Loiseau, “On the role of nonlinear correlations in reduced-order modelling,” *J. Fluid Mech.* **938**, A1 (2022).
- O. Hennigh, “Lat-Net: Compressing lattice Boltzmann flow simulations using deep neural networks,” *arXiv:1705.09036* [stat.ML] (2017).
- X. Chen, G. Yang, Q. Yao, Z. Nie, and Z. Jiang, “A compressed lattice Boltzmann method based on ConvLSTM and ResNet,” *Comput. Math. Appl.* **97**, 162–174 (2021).
- A. Corbetta, A. Gabbana, V. Gyrya, D. Livescu, J. Prins, and F. Toschi, “Toward learning lattice Boltzmann collision operators,” *Eur. Phys. J. E* **46**, 10 (2023).
- K. Fukami, K. Fukagata, and K. Taira, “Super-resolution reconstruction of turbulent flows with machine learning,” *J. Fluid Mech.* **870**, 106–120 (2019).
- K. Fukami, K. Fukagata, and K. Taira, “Super-resolution analysis via machine learning: A survey for fluid flows,” *Theor. Comput. Fluid Dyn.* **37**, 421–444 (2023).
- 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).
- 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).
- H. Gao, L. Sun, and J.-X. Wang, “Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels,” *Phys. Fluids* **33**, 073603 (2021).
- X. Chu, W. Guo, T. Wu, Y. Zhou, Y. Zhang, S. Cai, and G. Yang, “Flow reconstruction over a SUBOFF model based on LBM-generated data and physics-informed neural networks,” *Ocean Eng.* **308**, 118250 (2024).
- P. L. Bhatnagar, E. P. Gross, and M. Krook, “A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems,” *Phys. Rev.* **94**, 511–525 (1954).
- Y. H. Qian, D. D’Humières, and P. Lallemand, “Lattice BGK models for Navier–Stokes equation,” *Europhys. Lett.* **17**, 479 (1992).
- X. He and L.-S. Luo, “Lattice Boltzmann model for the incompressible Navier–Stokes equation,” *J. Stat. Phys.* **88**, 927–944 (1997).

- ³⁵D. R. Noble and J. R. Torczynski, "A lattice-Boltzmann method for partially saturated computational cells," *Int. J. Mod. Phys. C* **09**, 1189–1201 (1998).
- ³⁶X. Xiong, K. Lu, Z. Zhang, Z. Zeng, S. Zhou, R. Hu, and Z. Deng, "High-frequency flow field super-resolution via physics-informed hierarchical adaptive Fourier feature networks," *Phys. Fluids* **37**, 097111 (2025).
- ³⁷C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.* **38**, 295–307 (2016).
- ³⁸X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, edited by G. Gordon, D. Dunson, and M. Dudík (PMLR, 2011), Vol. 15, pp. 315–323.
- ³⁹C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Computer Vision—ECCV 2016*, edited by B. Leibe, J. Matas, N. Sebe, and M. Welling (Springer, Cham, 2016), pp. 391–407.
- ⁴⁰Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2018), pp. 2472–2481.
- ⁴¹K. 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, 2016), pp. 770–778.
- ⁴²W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), pp. 1874–1883.
- ⁴³R. Y. Luo, Q. Z. Li, G. B. Zu, Y. J. Huang, G. C. Yang, and Q. H. Yao, "A super-resolution lattice Boltzmann method based on convolutional neural network," *Chin. J. Theor. Appl. Mech.* **56**, 3612–3624 (2024).
- ⁴⁴B. N. Rajani, A. Kandasamy, and S. Majumdar, "Numerical simulation of laminar flow past a circular cylinder," *Appl. Math. Modell.* **33**, 1228–1247 (2009).
- ⁴⁵C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data* **6**, 60 (2019).
- ⁴⁶K. Fukami and K. Taira, "Single-snapshot machine learning for super-resolution of turbulence," *J. Fluid Mech.* **1001**, A32 (2024).
- ⁴⁷J. Page, "Super-resolution of turbulence with dynamics in the loss," *J. Fluid Mech.* **1002**, R3 (2025).
- ⁴⁸M. Morimoto, K. Fukami, K. Zhang, and K. Fukagata, "Generalization techniques of neural networks for fluid flow estimation," *Neural Comput. Appl.* **34**, 3647–3669 (2022).
- ⁴⁹Z. 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).
- ⁵⁰K. Fukami and K. Taira, "Observable-augmented manifold learning for multi-source turbulent flow data," *J. Fluid Mech.* **1010**, R4 (2025).