

---

# PEINR: A Physics-enhanced Implicit Neural Representation for High-Fidelity Flow Field Reconstruction

---

Liming Shen<sup>1,2</sup> Liang Deng<sup>3</sup> Chongke Bi<sup>4</sup> Yu Wang<sup>4</sup> Xinhai Chen<sup>1,2</sup> Yueqing Wang<sup>3</sup> Jie Liu<sup>1,2</sup>

## Abstract

Implicit neural representation (INR) excels in high-fidelity flow field reconstruction through flexible enhancement of numerical precision and grid resolution. However, its broader adoption faces two barriers: the absence of standardized benchmarks for flow reconstruction tasks, and the impractical grid independence assumption in real-world simulations. Current INR frameworks also struggle to resolve fine-scale structures and spatiotemporal dynamics, particularly under severe temporal-spatial data imbalance, where temporal sensitivity degrades significantly. Tackling these issues, we first introduce HFR-Bench, a 5.4 TB public large-scale CFD dataset with 33,600 unsteady 2D and 3D vector fields for reconstructing high-fidelity flow fields. We further present PEINR, a physics-enhanced INR framework, which is mainly composed of physical encoding and transformer-based spatiotemporal fuser (TransSTF). Physical encoding decouples temporal and spatial components through Gaussian temporal encoding, which can enhance high-dimensional features and nonlinear characteristics in temporal information, and localized spatial encoding, which can implement stencil-based discretization in the spatial dimension. TransSTF fuses both spatial and temporal information via transformer for capturing long-range temporal dependencies. Qualitative and quantitative experiments demonstrate that PEINR outperforms state-of-the-art INR-based methods in reconstruction quality. Code and dataset are released [here](#).

<sup>1</sup>Laboratory of Digitizing Software for Frontier Equipment, National University of Defense Technology, Changsha, China  
<sup>2</sup>National Key Laboratory of Parallel and Distributed Computing, National University of Defense Technology, Changsha, China  
<sup>3</sup>Computational Aerodynamics Institute, China Aerodynamics Research and Development Center, Mianyang, China  
<sup>4</sup>College of Intelligence and Computing, Tianjin University, Tianjin, China.  
Correspondence to: Liang Deng <dengliang11@nudt.edu.cn>.

## 1. Introduction

High-fidelity flow fields contain richer flow information and finer-scale flow structures, which play a crucial role in understanding the flow behaviors in various physical and natural phenomena. In computational fluid dynamics (CFD), increasing the grid-resolution and employing higher-order numerical schemes can typically generate high-fidelity flow fields, but the simulations require several days to complete and incur substantial computational costs. Recently, inspired by their success in computer vision, deep-learning-based super-resolution (SR) methods have increasingly been used to reconstruct high-fidelity flow fields from low-fidelity counterparts without repeatedly solving complex partial differential equations (Fukami et al., 2023).

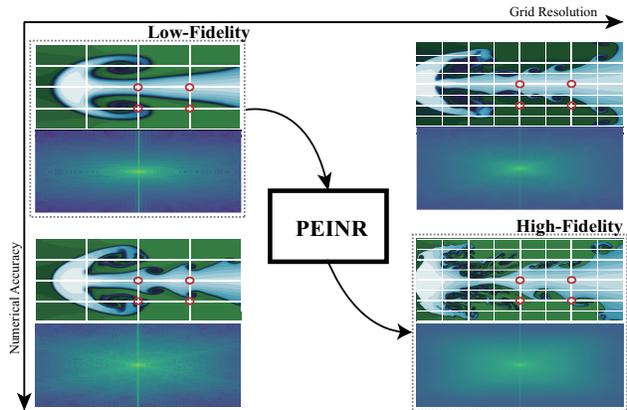


Figure 1. The red circles explain the error in the assumption of grid independence, showing that as the grid resolution (in the row direction) and numerical accuracy (in the column direction) increase, the attribute values at the same location differ and the high-frequency part becomes more pronounced in the corresponding frequency domain representations, which means that the structure of the flow field is more complex and refined.

Existing methods of high-fidelity flow field reconstruction based on convolutional neural networks (CNNs) (Fukami et al., 2021; Yousif et al., 2021; Xu et al., 2023; Hu et al., 2024; Shen et al., 2024b;a) are clearly confined to equally-spaced Cartesian meshes because the nodes of the computational grid are reinterpreted as pixels. Nevertheless, irregular meshes are widely used in the industry because they can

adequately delineate complex geometries and easily deal with localized regions that require different resolution, e.g. cells are small near walls to capture the boundary layer where flow transitions are sharp whereas they are large in the freestream where gradients are smooth. A grid can also be regarded as a graph and graph neural networks (GNNs) are reasonable candidates. However, in order to propagate information over long distances between nodes, many graph convolutional layers need to be stacked. Moreover, although pooling is a cheap operation in CNNs, it is rather challenging in GNNs (Grattarola et al., 2022; Kashefi et al., 2021).

The implicit neural representation (INR) frameworks (Han & Wang, 2023; Pan et al., 2023; Tang & Wang, 2024; Jiao et al., 2024) typically learn to represent a flow field as a continuous function which can map the spatiotemporal coordinates to their corresponding values, thus are highly adaptable for different mesh types and having emerged as a new mesh-agnostic paradigm for flow reconstruction. However, these INR-based methods still face several limitations in high-fidelity flow field reconstruction: (1) In the absence of standard benchmarks datasets, existing INR-based methods assume grid independence, where attribute values at the same location in the flow field remain unchanged across different mesh resolutions. However, in the case of real simulation data, this assumption does not hold. As shown in Figure 1, the red circles highlight the same locations in flow fields with varying mesh resolutions and numerical accuracies, where the corresponding density values clearly differ. (2) Due to the I/O bottleneck (Fajardo et al., 2018), only selective time steps can be saved. There is a significant disparity between temporal and spatial complexity. Existing methods overlook this gap by simply coupling space and time, thus failing to fully simulate the complex spatiotemporal dynamics. (3) The spectral bias issue of INR method (Xu, 2018; Rahaman et al., 2019) concentrate more on the low-frequency information, which can have deficiencies in capturing fine-scale structures of flow fields. As shown in Figure 1, the corresponding frequency domain representations illustrate that high-fidelity flow fields capturing small-scale flow features more effectively can exhibit a richer presence of high-frequency information.

In this paper, we first introduce HFR-Bench, a truly large-scale dataset containing 24,000 unsteady flow fields of three canonical two-dimensional (2D) flow problems, 1,600 unsteady flow fields of a three-dimensional (3D) flow problem with uniform Cartesian mesh, and 8,000 2D unsteady flow fields with non-uniform mesh, amounting to a total of 5.4 TB of data. For each problem, we discretize the domain into four different grid-resolution configurations, and perform a simulation at each grid-resolution using four different numerical-precision settings. Low- and high-fidelity simulations in each pair start from identical initial conditions and share the same physical parameters. The maximum

grid-resolution gap and the highest numerical-precision can reach up to 64 times and 7th-order, respectively.

Compared to high-fidelity simulation, the coarse discretization of the low-fidelity simulation introduces inaccuracies and truncation errors. As shown in Figure 2(a), we calculate the correction errors between low and high-fidelity flow fields. Effectively representing the error field is the target we strive to reach with the application of the INR network. We propose a physics-enhanced INR (PEINR) which can simultaneously enhancing the numerical-precision and grid-resolution of flow fields. PEINR consists of physical encoding and transformer-based spatiotemporal fuser (TransSTF). In physical encoding, as shown in Figure 2(b), spatial coordinates  $\vec{C}$  are expanded from neighbors, taking into account the nature of stencil discretization. Temporal information is expanded by Gaussian kernel encoding to addressing the issue of disproportionate spatiotemporal dimensions. As shown in Figure 2(c), TransSTF, a temporal-aware encoder based on the multi-head attention mechanism, aims to fuse both spatial and temporal information and capture long-range temporal dependencies. We also leverage spectral block (Patro et al., 2023), to capture the high frequency components of high fidelity flow fields. We perform qualitative and quantitative analyses, to experimentally demonstrate that our approach is well-suited for high-fidelity flow field reconstruction tasks.

The key contributions of our work are summarized as follows: (1) We release HFR-Bench, a truly large-scale CFD dataset including both uniform Cartesian and non-uniform meshes with 33,600 unsteady 2D and 3D vector fields, amounting to a total of 5.4 TB of data. For each simulation, data are provided at both low and high resolution, making the dataset suitable for the specific task of reconstructing high-fidelity flow fields. (2) We propose a novel physics-enhanced INR model for concurrently handling numerical-precision and grid-resolution enhancement for both uniform and non-uniform meshes. The physical encoding can alleviate the disparity between temporal and spatial complexities, capture the nonlinear characteristics of spatiotemporal dynamics and the stencil discretization of spatial dimensions. (3) We utilize the spectral block in TransSTF to alleviate the spectral bias on flow field learning. The result of ablation study can further illustrate that our method can alleviate the negative impact.

## 2. Related Work

This section discusses the related works of implicit neural representation, and super-resolution for flow field.

**Implicit neural representation.** Implicit neural representation (INR) is a powerful framework that seeks to train a neural network to parameterize a continuous, implicit func-

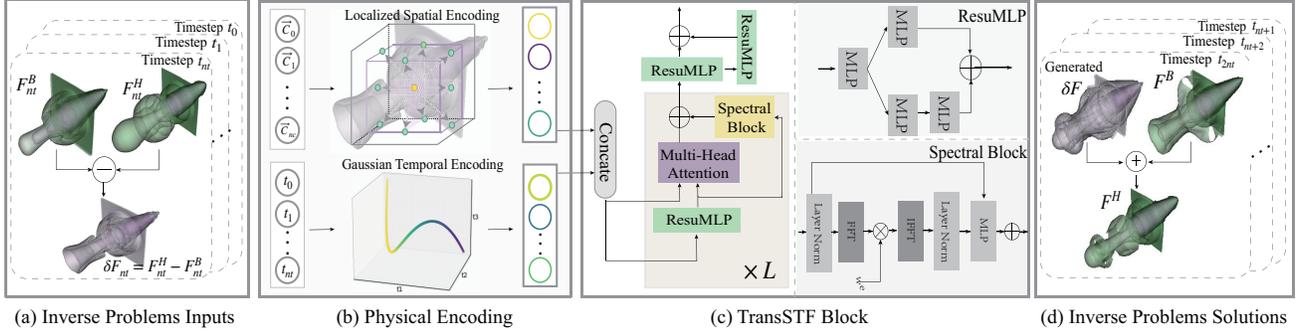


Figure 2. Overview of our method PEINR. (a) We first calculate the difference field between the high-fidelity flow field and low-fidelity counterpart. (b) In physical encoding, we leverage Gaussian coordinate encoding and localized encoding to handle temporal information and spatial coordinates. (c) The TransSTF block fuses temporal and spatial information and consists of ResuMLP, multi-head attention and spectral block. (d) The generated high-fidelity flow fields are obtained by combining difference fields and low-fidelity counterparts.

tion. This function maps a given spatiotemporal position, which consists of both spatial and temporal coordinates, to its corresponding value.

Recently, incorporating Fourier features (Mildenhall et al., 2021), periodic activations (Sitzmann et al., 2020), or multi-resolution Hash tables (Müller et al., 2022) has significantly improved the performance of INR for scene reconstruction (Jiang et al., 2020; Chabra et al., 2020) and shape modeling (Genova et al., 2020; Atzmon & Lipman, 2020). Recent work by de Vito et al. (de Vito et al., 2024) in implicit neural representation for accurate CFD flow field prediction demonstrates the potential of INR architectures for steady-state CFD simulations. However, their method primarily focuses on single-resolution flow fields with stationary boundary conditions and does not address the critical challenges of reconstructing unsteady, multi-resolution flow fields with spatiotemporal coupling – a key limitation our PEINR framework resolves through physical encoding and TransSTF.

PINNs (Karniadakis et al., 2021; Dong & Polak, 2024; Hosseini & Shiri, 2024) are an extended application of implicit neural representations in scientific computing. Nevertheless, our CFD data are solved based on discrete meshes, which inherently introduce numerical dissipation and truncation errors, deviating from the Navier-Stokes equations governing fluid flow. Forcing PINNs to simultaneously fit this data and satisfy the equation residuals may lead to network convergence issues or generate non-physical solutions (Farea et al., 2024).

**Super-resolution for flow field.** In the past several years, various deep learning-based methods have been applied to tackle single image super-resolution (SISR) tasks (Lu et al., 2022; Zhang et al., 2022; Chen et al., 2023). Inspired by their success in computer vision, SISR methods have increasingly been used to reconstruct high-fidelity flow fields

by simply replacing the red, green, and blue components with physical variables (Fukami et al., 2023). Among these works, CNN-based super-resolution models (Zhenglei et al., 2017; Fukami et al., 2019; 2021; Liu et al., 2020; Obiols-Sales et al., 2021; Gao et al., 2021) have been actively studied for a range of flows. To further improve the model performance, many works complicate their model by incorporating GAN (You et al., 2018; Zhiwen et al., 2019; Yousif et al., 2021; 2022; Yu et al., 2022; Han & Wang, 2019; 2020; Wurster et al., 2023; Hyojin et al., 2021), transformer (Wang et al., 2022; Xu et al., 2023; Hu et al., 2024; Shen et al., 2024b;a), for capturing fine-scale flow features. However, these methods are limited to flow field data on Cartesian grids.

Neural operators (Kovachki et al., 2023) leverage function space mappings and efficient spectral-domain computations to overcome the fixed-grid limitations of traditional methods. However, when high-frequency energy decays rapidly, neural operators may struggle to retain fine details compared to INR. Additionally, INR supports on-demand generation of localized regions, such as boundary layers or vortex structures, without requiring full-field computation, making it more adaptable and efficient.

Existing INR methods for flow field (Han & Wang, 2023; Pan et al., 2023; Tang & Wang, 2024; Jiao et al., 2024) only focus on grid-resolution enhancement, and PEINR is the first attempt to simultaneously enhance the grid-resolution and numerical-precision by leveraging transformer and physical-enhanced INR model.

### 3. Method

We propose a framework with an INR-based model for high-fidelity flow field reconstruction. The framework consists of 4 components: inverse problem input encoding, physi-

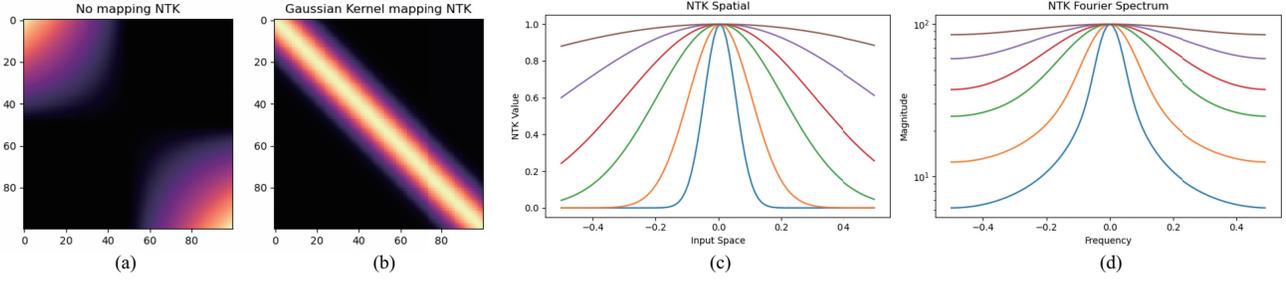


Figure 3. (a) The NTK without mapping exhibits irregular patterns, indicating dependency on absolute time positions and leading to inconsistent predictions. (b) Applying the Gaussian kernel mapping results in a structured diagonal pattern, ensuring translation invariance and improved generalization. (c) The NTK spatial decay plot shows how kernel values diminish across input space, with smaller  $\sigma$  emphasizing local interactions and larger  $\sigma$  capturing broader dependencies. (d) The NTK Fourier spectrum demonstrates that larger  $\sigma$  values allow high-frequency variations, while smaller  $\sigma$  favor smooth, low-frequency behaviors.

cal encoding, TransSTF, and inverse problem solutions, as shown in Figure 2. Our method allows approximating the accurate variables (density  $\rho$ , the components  $u$ ,  $v$  and  $w$  of vector variable velocity  $\mathbf{V}$ ) of high-fidelity flow field by learning its time information and the dependency of spatial information. We detail all framework components and learning schemes in the following subsections.

### 3.1. Inverse Problem Input Encoding and Solutions

The inverse problem refers to reconstructing high-fidelity physical fields from sparse data (Chen et al., 2024). The CFD simulation flow field is closely related to the numerical accuracy and grid resolution, and these two factors influence each other. Finer grids can better capture flow details and reduce discretization errors, which can lead to discrepancies in flow field property values at the same location. To enhance the applicability of INR, we chose to use INR to learn the error field  $\delta F$  between high-fidelity and low-fidelity flow fields.

The process is as follows: to establish the connection between the low-fidelity flow field  $F_{t_i}^L$  at timestep  $t_i$  and its high-fidelity counterpart  $F_{t_i}^H$ , we need to address the inaccuracies and truncation errors introduced by the coarse discretization in low-fidelity simulations. First, we upscale the low-fidelity flow field to match the spatial resolution of the high-fidelity field using BI (bicubic interpolation) method. Although traditional upscaling methods or transferring existing mature super-resolution models could be chosen, we opted for the traditional BI method due to the diversity of flow field grid types. BI method estimates the value at each point by considering the surrounding  $4 \times 4$  grid of known values and applying cubic interpolation along both axes, resulting in a smoother and more accurate up-scaled flow field compared to simpler methods like bilinear interpolation. The resulting up-scaled flow field, denoted as  $F_{t_i}^B$ , provides a better approximation of the high-fidelity

field but still contains residual errors.

As shown in Figure 2 (a), to quantify these discrepancies, we compute the error field  $\delta F_{t_i} = F_{t_i}^H - F_{t_i}^B$ , which captures the missing fine-grained details. We obtain  $\delta F_{t_i}$  through the BI method, which allows us to efficiently handle the discrepancies between the low-fidelity and high-fidelity flow fields, thus laying the foundation for subsequent error learning and precise reconstruction. The goal of the PEINR network is to effectively learn and represent the error field  $\delta F_{t_i}$ , enabling the accurate reconstruction of high-fidelity flow fields from low-fidelity counterparts. As shown in Figure 2 (d), to obtain the inverse problems solutions, we use the error field generated by INR model along with the low-fidelity flow field to reconstruct the high-fidelity flow field.

### 3.2. Physical Encoding

To accurately capture the spatial and temporal dependencies in flow field representations, we introduce specialized encoding techniques for both spatial and temporal information.

#### 3.2.1. SPATIAL DISCRETIZATION

Spatial discretization in computations typically involves stencil-based methods, whether using finite volume, finite difference, or finite element methods. These approaches rely on information from grid cells and their neighboring cells to construct the discretized equations. In most conventional INR methods, the network learns to represent a flow field as a continuous function with the input of dependent coordinates  $\vec{C}$  and temporal information  $t$ , ignoring the importance of spatial dependency.

To address this, after randomly sampling a sufficient number of points, we expand the input spatial coordinates to include not only the original coordinates but also those of nearby

neighboring points:

$$\Psi(\vec{C}) = \left[ \vec{C}, \vec{C}_{\text{neighbors}} \right] \quad (1)$$

This localized encoding aligns with stencil-based computation paradigm of CFD, where numerical solutions inherently depend on local neighborhood interactions (e.g., finite volume discretization). By embedding this locality into our spatial localization encoding, PEINR explicitly leverages the template computation logic to resolve fine-scale flow structures. For instance, in a 2D case, we consider the nearest 4 points (up, down, left, right), transforming the input from a single coordinate tensor of shape (batchsize, 2) to an augmented tensor of shape (batchsize, 5, 2). This design explicitly encodes the local spatial correlations required for solving discretized Navier-Stokes equations, where the solution of each grid point depends on physical quantities of its neighbors.

Spatial discretization allows the model to capture spatial derivatives more effectively, reflecting the underlying properties of partial differential equations (PDEs) that govern flow field computations. By considering spatial derivatives, the model can better account for the local gradients and variations in the flow field, enhancing its ability to represent fine-scale structures and accurately approximate the dynamics dictated by the governing PDEs, as shown in Figure 2 (b).

### 3.2.2. NONLINEAR TEMPORAL ENCODING

To effectively model the nonlinear temporal dynamics in flow fields, we employ a two-step approach that combines the Gaussian Radial Basis Function (RBF) kernel and Kernel Principal Component Analysis (Kernel PCA). This method transforms the one-dimensional temporal input into a high-dimensional, nonlinear feature space and then reduces the dimensionality while preserving essential temporal features.

**Step 1: Temporal Encoding Using RBF Kernel.** The RBF kernel maps the one-dimensional temporal input  $t$  into a high-dimensional feature space where nonlinear temporal patterns can be better represented. The RBF kernel function is defined as:

$$K(t, t') = \exp\left(-\frac{|t - t'|^2}{2\sigma^2}\right), \quad (2)$$

where  $t$  and  $t'$  are two time instances.  $\sigma$  is a hyperparameter controlling the kernel's width, determining how quickly the similarity between  $t$  and  $t'$  decays with their distance.

Given a set of temporal data points  $\{t_1, t_2, \dots, t_N\}$ , we construct the RBF kernel matrix  $K$  of dimensions  $N \times N$ , encoding pairwise temporal relationships. This transformation ensures translation invariance by making the kernel

solely dependent on relative differences:

$$K(t_i, t_j) = K(t_i - t_j). \quad (3)$$

This property allows the INR model to maintain consistency across different time steps, improving long-term reconstruction accuracy.

To analyze the impact of the RBF kernel on model stability and generalization, we apply Neural Tangent Kernel (NTK) theory, which describes the training evolution of infinitely wide neural networks and is essential for convergence analysis. Traditional NTKs derived from MLPs lack translation invariance, making them unsuitable for temporal modeling. By incorporating the RBF kernel, we convert the NTK into a stationary kernel that depends on relative time differences, improving stability, especially in solving time-dependent PDEs. As shown in Figure 3, the RBF-enhanced NTK exhibits structured patterns that better capture temporal dependencies, promoting smoothness and consistency over time, which enhances generalization and robustness in dynamic environments. The spectral properties of the NTK reveal that adjusting the kernel bandwidth  $\sigma$  allows for flexible adaptation to various time scales, balancing local and global temporal interactions.

**Step 2: Extracting Temporal Components with Kernel PCA.** Kernel PCA is applied to extract significant temporal features from the kernel matrix by first centering the matrix to ensure zero-mean data. This is achieved using the equation:

$$\tilde{K} = K - \mathbf{1}_N K - K \mathbf{1}_N + \mathbf{1}_N K \mathbf{1}_N \quad (4)$$

where  $\mathbf{1}_N$  is an  $N \times N$  matrix with all elements equal to  $1/N$ .

Next, eigenvalue decomposition of the centered kernel matrix  $\tilde{K}$  yields eigenvalues and eigenvectors, from which the top  $m$  eigenvectors are selected to form a low-dimensional representation. Temporal inputs are projected into this subspace to obtain their encoded representations:

$$\Phi(t_i) = V_m^\top K(t_i) \quad (5)$$

where  $\Phi(t)$  is the nonlinear feature representation of  $t$  in the reduced space,  $V_m$  is the matrix of the top  $m$  eigenvectors, and  $K(t_i)$  is the similarity vector of time point  $t_i$  with all other points (i.e., the  $i$ -th column of the kernel matrix).

The combination of the RBF kernel and Kernel PCA captures complex temporal relationships, retaining key features through dimensionality reduction, ensuring smooth transitions in representation, and enhancing the INR model's efficiency and stability in learning nonlinear temporal dynamics.

The encoded temporal features  $\Phi(t)$  are concatenated with the localized encoding spatial coordinates  $\Psi(\vec{C})$  and fed

into the INR model. The INR framework approximates the error field  $\delta F(x, y, z, t)$  by learning a continuous mapping:

$$\delta F(x, y, z, t) \approx \mathcal{N}(\Phi(t), \Psi(\vec{C}), \theta), \quad (6)$$

where  $\mathcal{N}$  is the neural network parameterized by  $\theta$ . This approach enables the model to leverage the enriched temporal features for improved reconstruction accuracy.

### 3.3. TransSTF

As shown in Figure 2 (c), TransSTF combines spatial and temporal information to generate complex embeddings using higher-order features. It consists of multi-head attention (MHA) layers, ResuMLPs, and spectral blocks. The MHA layer with query ( $Q$ ), key ( $K$ ), and value ( $V$ ) inputs aims to refine feature maps, where  $Q$ ,  $K$ , and  $V$  come from a mixture of spatial-temporal data and ResuMLP-extracted features. The output matrix of MHA ( $\mathcal{M}$ ) is shown as follows:

$$\begin{aligned} \mathcal{M}(Q, K, V) &= \text{Concat}(\text{att}_1, \dots, \text{att}_h) W^O \\ \text{where att}_i &= \text{softmax} \left( \frac{Q_i W_i^Q K_i^T W_i^K}{\sqrt{d_k}} \right) V_i W_i^V \end{aligned} \quad (7)$$

**ResuMLP.** The ResuMLP (Han & Wang, 2022) improves gradient propagation by adding depth and complexity to the network. It includes residual blocks and a SIREN (Sitzmann et al., 2020) activation function ( $\sin(\omega x)$ ), with  $\omega$  set to 30, as recommended (Sitzmann et al., 2020). The output is scaled to the  $[-1, 1]$  range, suitable for sinusoidal activations.

**Spectral Block.** The spectral block (Patro et al., 2023) captures different frequency components and reduces spectral bias. It includes a spectral gating network with FFT for converting to spectral space, weighted gating to adjust frequency component weights, and an inverse FFT (IFFT) to return the signal to physical space, enhancing the capture of high-frequency features like edges.

## 4. Experiment

**Dataset.** We generate a truly large-scale dataset HFR-Bench, containing 2D and 3D unsteady flow fields including both uniform Cartesian and non-uniform meshes in different high-order weighted essentially non-oscillatory (WENO) schemes, amounting to a total of 5.4 TB of data. In uniform Cartesian meshes, we choose the Rayleigh-Taylor instability (RT), Riemann (RM), and Forward Facing Step (FFS) problems for the 2D flow fields, and the shock-longitudinal vortex interaction (SV) case in 3D. In non-uniform structured meshes, we simulate the problem of a flow past a cylinder (Cylinder). The discrepancy in grid-resolution and

numerical-precision between  $F^L$  and  $F^H$  data can be described by the upscaling factor  $\alpha$  and the improvement factor  $\beta$ :

$$\alpha = ST_{HF}/ST_{LF}, \quad \beta = SP_{HF}/SP_{LF}, \quad (8)$$

where  $ST$  denotes the number of grid points and  $SP$  denotes the number of points in the stencil.

**Baselines.** We compare our method with three classic methods: (1) Bicubic interpolation (BI) is a common traditional image processing algorithm based on interpolation. (2) NIF (23’JMLR) (Pan et al., 2023) consists of two modified MLPs: ShapeNet, which isolates and represents spatial complexity, and ParameterNet, which represents other inputs, such as time. (3) CoordNet (23’TVCG) (Han & Wang, 2023) is coordinate-based and leverages an encoder-decoder based INR to learn a mapping from coordinates to values.

**Implementation Details.** All experiments are conducted on a single NVIDIA A100 80GB GPU. The batch size (BS), memory (MM) and experimental variable and time per epoch are listed in Table 1.

Our method and all the baseline methods are trained with the MSE loss in 2000 epochs and every method including ours can converge within 1000 epochs. Learning ratio is set to  $1e-5$  and decrease after 20 epochs if there is no loss degradation and we adopt the AdamW optimizer for optimization. In spatial discretization, for 2D cases, we consider the nearest 4 points, and for 3D cases, the nearest 9 points. In temporal nonlinear encoding, we set the  $\sigma$  as 10 with time steps normalized to  $[0, 1]$ . During design, we set the number of residual layers of a ResuMLP to 10 and the max number of neurons in a ResuMLP is 64. Experiments are conducted within a range of  $\pm 10\%$  around the optimal hyperparameters, using the inference results as the standard to confirm that the hyperparameters are indeed optimal. For the other models compared in the table, we utilized their original code and conducted experiments by only varying the input length.

Experiments are conducted on a single channel for constructing  $\rho$  in flow fields, and two (three) channels for constructing velocity fields in 2D (3D). For uniform Cartesian meshes, flow fields from timesteps 460 to 480 (out of 500) are used for training, with the final 20 steps as test samples. Results for uniform meshes (FFS, RM, RT, SV) are shown for step 500. For non-uniform meshes, training samples are taken from steps 400 to 500 at intervals of 5, with the remaining non-multiples of 5 used for testing. Results for non-uniform meshes are presented for step 494.

### 4.1. Quantitative Comparison

We take MSE (Mean Squared Error) as the loss function and report PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), CORR (Correlation Coefficient) and

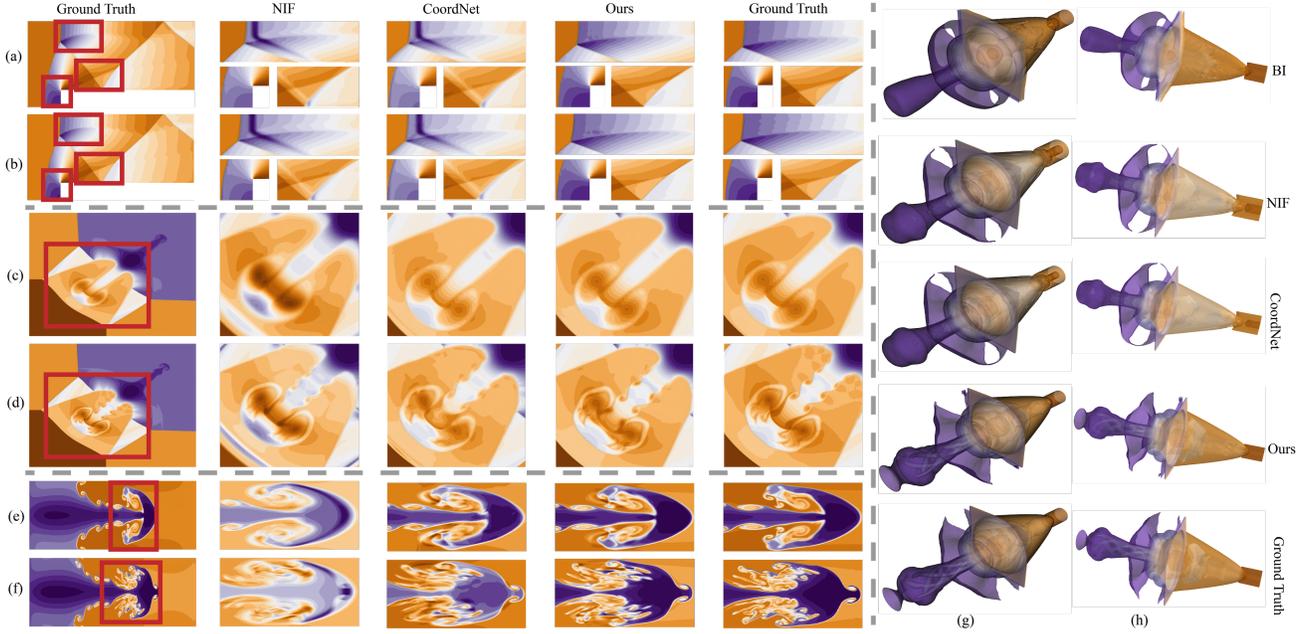


Figure 4. Comparison results of the variable  $\rho$  flow field of uniform Cartesian meshes, given by NIF, CoordNet and our method in the 2D (upscaling  $\alpha = 16$ ) and 3D datasets (upscaling  $\alpha = 64$ ). Results of FFS datasets in (a) WENO3 ( $\beta = 1$ ), (b) from WENO3 to WENO7 with  $\beta = 2.6$ . Results of RM datasets in (c) WENO3 ( $\beta = 1$ ), (d) from WENO3 to WENO5 with  $\beta = 1.4$ . Results of RT datasets in (e) WENO3 ( $\beta = 1$ ), (f) from WENO3 to WENO5 with  $\beta = 1.4$ . (g) (h) Results of SV datasets from WENO3 to WENO5 with  $\beta = 1.4$  from different viewpoints.

DD (Dissipation difference) results as the evaluation metrics. The dissipation operator is used to assess the model performance in terms of capturing the discontinuities and rotating features with lower values indicating better performance with lower values indicating better performance.

The quantitative results are shown in Table 2, which quantitatively compares PEINR with different grid-resolution and numerical-precision against the results generated by baselines using data-level metrics (higher values are better except for DD). In Table 2, all experiments are one-channel, conducted to reconstruct the variable  $\rho$  of flow fields. PEINR can generally outperform the baselines, except for the slightly inferior PSNR with respect to CoordNet (Han & Wang, 2023) in the RM dataset. The observed PSNR difference stems from PEINR’s design prioritizing physical accuracy over numerical optimization for discontinuous flows. Riemann problems involve strong discontinuities (shocks, contact surfaces) where PEINR’s localized spatial encoding intentionally preserves sharp gradients, making it more sensitive to errors in these regions. While this approach leads to slightly lower PSNR (a mean-squared-error metric favoring smoothness), it avoids the physical distortions visible in CoordNet’s results (Figure 4(c)(d)) where large errors near discontinuities appear.

As shown in 5, we also present the four evaluation met-

rics for Table 2 (a) across all extrapolated timesteps. The results clearly demonstrate that PEINR exhibits an absolute advantage in generalization performance.

**Ablation Study** To evaluate the effectiveness of importance sampling and spectral block of PEINR, we compare with two baseline methods PEINR<sup>1</sup> and PEINR<sup>2</sup>. PEINR<sup>1</sup> removes both spectral block and the physical encoding, while PEINR<sup>2</sup> only removes the physical encoding. As shown in Table 2, PEINR outperforms PEINR<sup>2</sup> in the metric DD in all cases, which indicates that our physical encoding method can preserve more physical peculiarity. The superiority of PEINR<sup>2</sup> compared with PEINR<sup>1</sup> highlights that the added spectral block can also help the model to perform better.

## 4.2. Qualitative Comparison

The qualitative evaluation provides several contour plots and streamline rendering results. Figure 4 displays synthesized flow fields of density  $\rho$  of FFS datasets in (a) (b), RM datasets in (c) (d), RT datasets in (e) (f) and 3D SV datasets in (g) (h) with different upscaling factors  $\alpha$  and improvement factors  $\beta$ . To facilitate analysis, we normalized all results to the interval  $[-1, 1]$  for ease of observation. The areas of interest to domain experts are marked with red squares in the first column, and we zoom in on these areas in the following columns. BI can not capture complex fluids

Table 1. The details of each data set. GR, NP, BS and MM denote grid resolution, numerical precision, batch size and memory, respectively.

dataset	Variable	Input		Output		Train			Inference
		GR	NP	GR	NP	BS	Time(s)	MM(GB)	Time(s)
RT	$\rho, u, v$	120×480	WENO3	480×1920	WENO3	8000	10	22.68	16.71
					WENO5				
RM	$\rho, u, v$	200×200	WENO3	800×800	WENO3	8000	11	10.53	13.59
					WENO5				
FFS	$\rho, u, v$	36×12	WENO3	144×48	WENO3	8000	11	17.29	15.4
		180×48		720×192	WENO7				
Cylinder	$u, v$	51×381	WENO3	101×761	WENO3	8000	19	5.11	9.59
					WENO5				
SV	$\rho, u, v, w$	96×32×32	WENO3	384×128×128	WENO5	36000	23	68.45	75.31

Table 2. PSNR, SSIM×10<sup>2</sup>, CORR×10<sup>2</sup> and DD×10<sup>5</sup> values with different upscaling factors  $\alpha$  and improvement factors  $\beta$ . The best values are highlighted in bold.

Dataset	(a) RT $\alpha=16 \beta=1$				(d) RM $\alpha=16 \beta=1$				(g) FFS $\alpha=16 \beta=1$			
Metric	PSNR	SSIM	CORR	DD	PSNR	SSIM	CORR	DD	PSNR	SSIM	CORR	DD
NIF	17.53	87.70	94.04	22.46	24.83	92.85	98.33	26.45	35.09	86.44	95.78	2991.72
CoordNet	23.33	93.17	98.40	10.25	<b>45.34</b>	99.67	<b>99.98</b>	21.80	42.71	94.88	98.61	98.46
PEINR <sup>1</sup>	24.82	94.51	98.86	9.31	34.16	98.70	99.80	4.93	50	96.47	98.96	72.92
PEINR <sup>2</sup>	26.91	95.85	99.28	8.17	37.10	98.96	99.90	3.28	70.03	99.80	<b>99.98</b>	15.93
PEINR	<b>26.96</b>	<b>95.91</b>	<b>99.31</b>	<b>7.76</b>	44.22	<b>99.76</b>	<b>99.98</b>	<b>2.51</b>	<b>74.15</b>	<b>99.84</b>	<b>99.98</b>	<b>4.61</b>
Dataset	(b) RT $\alpha=16 \beta=1.4$				(e) RM $\alpha=16 \beta=1.4$				(h) FFS $\alpha=16 \beta=2.6$			
Metric	PSNR	SSIM	CORR	DD	PSNR	SSIM	CORR	DD	PSNR	SSIM	CORR	DD×10 <sup>4</sup>
NIF	15.15	83.81	89.98	85.02	23.93	90.002	97.95	17.46	48.43	95.64	98.81	193.82
CoordNet	20.73	89.20	97.16	71.38	<b>37.98</b>	98.46	<b>99.91</b>	8.78	41.16	93.92	98.41	246.5
PEINR <sup>1</sup>	21.45	89.65	97.61	26.32	33.27	97.57	99.75	7.20	37.14	96.50	99.10	314.26
PEINR <sup>2</sup>	22.40	91.01	97.88	24.58	37.55	<b>98.66</b>	99.90	3.71	51.10	99.40	99.92	58.97
PEINR	<b>22.46</b>	<b>91.07</b>	<b>97.93</b>	<b>22.09</b>	37.81	98.62	<b>99.91</b>	<b>3.18</b>	<b>70.91</b>	<b>99.46</b>	<b>99.96</b>	<b>25.38</b>
Dataset	(c) Cylinder $\alpha=4 \beta=1$				(f) Cylinder $\alpha=4 \beta=1.4$				(i) SV $\alpha=64 \beta=1.4$			
Metric	PSNR	SSIM	CORR	DD	PSNR	SSIM	CORR	DD	PSNR	SSIM	CORR	DD×10 <sup>4</sup>
NIF	16.28	80.12	82.88	131.58	19.08	78.20	84.80	286.9	27.47	89.67	99.21	20.75
CoordNet	21.80	84.32	92.20	12.76	19.31	85.09	90.30	172.7	27.34	89.57	99.19	21.96
PEINR <sup>1</sup>	39.15	98.52	99.85	11.53	35.7	98.88	99.86	49.6	42.21	99.12	99.97	9.91
PEINR <sup>2</sup>	39.47	98.52	99.85	7.63	38.9	98.89	99.81	11.25	43.55	99.29	99.98	11.22
PEINR	<b>40.04</b>	<b>99.10</b>	<b>99.89</b>	<b>2.35</b>	<b>40.34</b>	<b>99.28</b>	<b>99.88</b>	<b>1.27</b>	<b>45.26</b>	<b>99.50</b>	<b>99.98</b>	<b>4.21</b>

due to its simple interpolation, and the low-fidelity and high-fidelity data are very different in near-real circumstances, as shown in Figure 4 (g) (h). NIF tends to blur the turbulence structures, resulting in poor reconstructions of the small-scale ground-truth features, regardless of  $\alpha$  or  $\beta$ . PEINR can efficiently handle the simultaneous enhancement of the numerical-precision and grid-resolution.

In Figure 6, we compare streamline rendering results of the synthesized vector fields generated by baselines. Figure 6 (a) and (b) display 48 streamlines generated from different seeds in the SV data set. While NIF and CoordNet can capture more details compared with BI but the tail of vortex obtained are diverging and they fail to produce more complex flow details. In contrast, our approach not only maintains a high level of detail but also ensures that the complex behaviors of the flow, including vortex dynamics and the underlying fluid structure, are consistently preserved

throughout the simulation. Figure 7 displays the comparison results of Cylinder dataset in non-uniform structured meshes, and it can illustrate that our method achieves better results on both uniform Cartesian grid data and non-uniform structured data compared to other baselines.

## 5. Conclusion

We introduce a large-scale comprehensive flow simulation dataset covering 5 canonical flow problems of 2D and 3D in both uniform Cartesian and non-uniform meshes. For each problem, results were obtained using four grid-resolution and four numerical-precision settings. In total, the dataset contains 33,600 vector fields, resulting in approximately 5.4 TB of data. Using this dataset, we creatively propose the coordinate-based INR method with the attention mechanism and physical encoding, which achieves satisfactory results

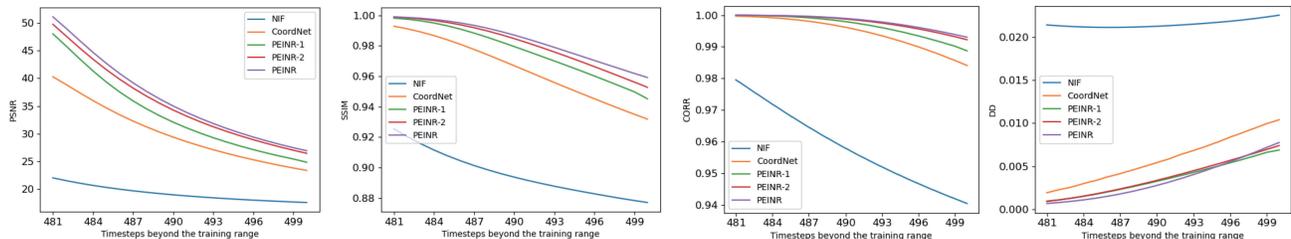


Figure 5. Comparison of model generalization performance on unseen timesteps beyond the training range. The PEINR method consistently achieves higher PSNR, SSIM and CORR values while maintaining lower DD compared to other methods, demonstrating its superior generalization capability in long-term flow field predictions.

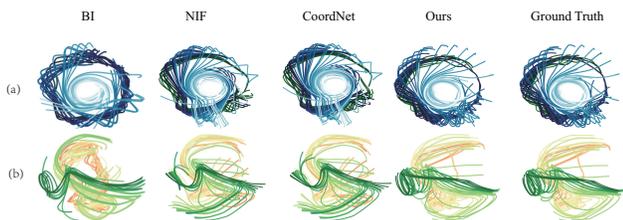


Figure 6. Comparison of streamline rendering results of three-channel experiments of SV dataset. Results given by BI, NIF and CoordNet are the inferred results (i.e., the networks do not see these vector fields during training). (a) and (b) display 48 streamlines from different seeds.

in the field of high-fidelity flow fields reconstruction, so that domain experts can observe high-fidelity flow fields with less storage cost. However, the PEINR model is, thus far, tailored and appraised for particular flow scenarios. In the future, we would like to explore meta-learning approaches to enhance the performance on out-of-distribution samples by considering the relationship among different flow fields.

## Impact Statement

This work introduces HFR-Bench, the first large-scale benchmark dataset for high-fidelity flow field reconstruction. By integrating physics-based encoding with transformer architectures, our proposed PEINR framework significantly improves the accuracy and robustness of implicit neural representations. These contributions pave the way for more generalizable and physically consistent AI solutions in fluid simulation tasks.

## Acknowledgment

This work was supported by the the National Key Research and Development Program of China (2023YFA1011704), and the National Key Research and Development Program of China (2021YFB0300101).

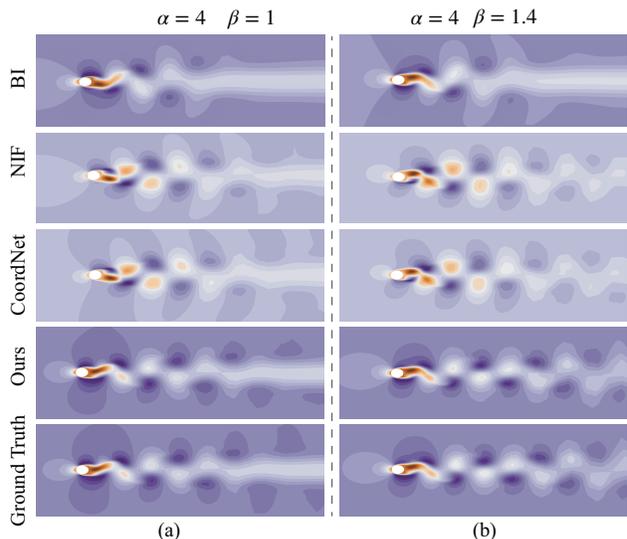


Figure 7. Comparison results of the variable  $u$  flow field with 25 uniformly distributed contours (upscaling  $\alpha = 4$ ) given by BI, NIF, CoordNet and our method in the Cylinder dataset of non-uniform structured meshes. Results in (a) WENO3 with  $\beta = 1$ , and (b) from WENO3 to WENO5 ( $\beta = 1.4$ ).

## References

- Atzmon, M. and Lipman, Y. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- Chabra, R., Lenssen, J. E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., and Newcombe, R. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pp. 608–625. Springer, 2020.
- Chen, X., Wang, X., Zhou, J., Qiao, Y., and Dong, C. Activating more pixels in image super-resolution transformer.

- In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22367–22377, June 2023.
- Chen, Y., Wu, R., Liu, Y., and Zhu, C. Hoin: High-order implicit neural representations. *arXiv preprint arXiv:2404.14674*, 2024.
- de Vito, L., Pinnau, N., and Dey, S. Implicit neural representation for accurate cfd flow field prediction. *arXiv preprint arXiv:2408.06486*, 2024.
- Dong, Z. and Polak, P. Cp-pins: Data-driven change-points detection in pdes using online optimized physics-informed neural networks. In *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, pp. 90–97. IEEE, 2024.
- Fajardo, C. A., Reyes, O. M., and Castillo, J. Reducing the i/o bottleneck by a compression strategy. *Engineering Letters*, 26(2), 2018.
- Farea, A., Yli-Harja, O., and Emmert-Streib, F. Understanding physics-informed neural networks: techniques, applications, trends, and challenges. *AI*, 5(3):1534–1557, 2024.
- Fukami, K., Fukagata, K., and Taira, K. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019.
- Fukami, K., Fukagata, K., and Taira, K. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *Journal of Fluid Mechanics*, 909:A9, 2021.
- Fukami, K., Fukagata, K., and Taira, K. Super-resolution analysis via machine learning: a survey for fluid flows. *Theoretical and Computational Fluid Dynamics*, pp. 1–24, 2023.
- Gao, H., Sun, L., and Wang, J.-X. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, 33(7), 2021.
- Genova, K., Cole, F., Sud, A., Sarna, A., and Funkhouser, T. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- Grattarola, D., Zambon, D., Bianchi, F. M., and Alippi, C. Understanding pooling in graph neural networks. *IEEE transactions on neural networks and learning systems*, 35(2):2708–2718, 2022.
- Han, J. and Wang, C. TSR-TVD: Temporal super-resolution for time-varying data analysis and visualization. *IEEE transactions on visualization and computer graphics*, 26(1):205–215, 2019.
- Han, J. and Wang, C. SSR-TVD: Spatial super-resolution for time-varying data analysis and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(6):2445–2456, 2020.
- Han, J. and Wang, C. Coordnet: Data generation and visualization generation for time-varying volumes via a coordinate-based neural network. *IEEE Transactions on Visualization and Computer Graphics*, 2022.
- Han, J. and Wang, C. CoordNet: Data generation and visualization generation for time-varying volumes via a coordinate-based neural network. *IEEE Transactions on Visualization and Computer Graphics*, 29(12):4951–4963, 2023.
- Hosseini, M. Y. and Shiri, Y. Flow field reconstruction from sparse sensor measurements with physics-informed neural networks. *Physics of Fluids*, 36(7), 2024.
- Hu, B., Yin, Z., Hamrani, A., Leon, A., and McDaniel, D. Super-resolution-assisted rapid high-fidelity cfd modeling of data centers. *Building and Environment*, 247:111036, 2024.
- Hyojin, K., Junhyuk, K., Sungjin, W., and Changhoon, L. Unsupervised deep learning for super-resolution reconstruction of turbulence. *Journal of Fluid Mechanics*, 910:A29, 2021.
- Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T., et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6001–6010, 2020.
- Jiao, C., Bi, C., and Yang, L. Ffeinr: flow feature-enhanced implicit neural representation for spatiotemporal super-resolution. *Journal of Visualization*, pp. 1–17, 2024.
- Jing, S., YongTao, Z., and ChiWang, S. Resolution of high order WENO schemes for complicated flow structures. *Journal of Computational Physics*, 186(2):690–696, 2003.
- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- Kashefi, A., Rempe, D., and Guibas, L. J. A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries. *Physics of Fluids*, 33(2), 2021.
- Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural

- operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- Liu, B., Tang, J., Huang, H., and Lu, X.-Y. Deep learning methods for super-resolution reconstruction of turbulent flows. *Physics of Fluids*, 32(2), 2020.
- Lu, Z., Li, J., Liu, H., Huang, C., Zhang, L., and Zeng, T. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 457–466, 2022.
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- Müller, T., Evans, A., Schied, C., and Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15, 2022.
- Obiols-Sales, O., Vishnu, A., Malaya, N. P., and Chandramowlishwaran, A. Surfnet: Super-resolution of turbulent flows with transfer learning using small datasets. In *2021 30th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pp. 331–344. IEEE, 2021.
- Pan, S., Brunton, S. L., and Kutz, J. N. Neural implicit flow: a mesh-agnostic dimensionality reduction paradigm of spatio-temporal data. *Journal of Machine Learning Research*, 24(41):1–60, 2023.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- Patro, B. N., Namboodiri, V. P., and Agneeswaran, V. S. Spectformer: Frequency and attention is what you need in a vision transformer. *arXiv preprint arXiv:2304.06446*, 2023.
- Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.
- Shen, L., Deng, L., Liu, X., Wang, Y., Chen, X., and Liu, J. A generative adversarial network based on an efficient transformer for high-fidelity flow field reconstruction. *Physics of Fluids*, 36(7), 2024a.
- Shen, L., Deng, L., Wang, Y., Zhang, J., and Liu, J. Pc-sagan: a physics-constrained generative network based on self-attention for high-fidelity flow field reconstruction. *Journal of Visualization*, pp. 1–16, 2024b.
- Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.
- Tang, K. and Wang, C. Stsr-inr: Spatiotemporal super-resolution for multivariate time-varying volumetric data via implicit neural representation. *Computers & Graphics*, 119:103874, 2024.
- Wang, X., Zhu, S., Guo, Y., Han, P., Wang, Y., Wei, Z., and Jin, X. Transflownet: A physics-constrained transformer framework for spatio-temporal super-resolution of flow simulations. *Journal of Computational Science*, 65:101906, 2022.
- Wurster, S. W., Guo, H., Shen, H.-W., Peterka, T., and Xu, J. Deep hierarchical super resolution for scientific data. *IEEE Transactions on Visualization and Computer Graphics*, 29(12):5483–5495, 2023.
- Xu, Q., Zhuang, Z., Pan, Y., and Wen, B. Super-resolution reconstruction of turbulent flows with a transformer-based deep learning framework. *Physics of Fluids*, 35(5), 2023.
- Xu, Z. J. Understanding training and generalization in deep learning by fourier analysis. *arXiv preprint arXiv:1808.04295*, 2018.
- You, X., Erik, F., Mengyu, C., and Nils, T. TempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow. *ACM Trans Graph*, 37(4), jul 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201304. URL <https://doi.org/10.1145/3197517.3201304>.
- Yousif, M. Z., Yu, L., and Lim, H.-C. High-fidelity reconstruction of turbulent flow from spatially limited data using enhanced super-resolution generative adversarial network. *Physics of Fluids*, 33(12), dec 2021. ISSN 1089-7666. doi: 10.1063/5.0066077. URL <http://dx.doi.org/10.1063/5.0066077>.
- Yousif, M. Z., Yu, L., and Lim, H.-C. Super-resolution reconstruction of turbulent flow fields at various reynolds numbers based on generative adversarial networks. *Physics of Fluids*, 34(1), jan 2022. ISSN 1089-7666. doi: 10.1063/5.0074724. URL <http://dx.doi.org/10.1063/5.0074724>.
- Yu, L., Yousif, M. Z., Zhang, M., Hoyas, S., Vinuesa, R., and Lim, H.-C. Three-dimensional ESRGAN for super-resolution reconstruction of turbulent flows with tricubic interpolation-based transfer learning. *Physics of Fluids*, 34(12), 2022.

Zhang, X., Zeng, H., Guo, S., and Zhang, L. Efficient long-range attention network for image super-resolution. In *European Conference on Computer Vision*, pp. 649–667. Springer, 2022.

Zhenglei, Z., Yule, H., Qirui, W., Guangxiang, C., Jiawei, L., Yubo, T., and Hai, L. Volume upscaling with convolutional neural networks. In *Proceedings of the Computer Graphics International Conference, CGI 17*. Association for Computing Machinery, 2017. ISBN 9781450352284. doi: 10.1145/3095140.3095178. URL <https://doi.org/10.1145/3095140.3095178>.

Zhiwen, D., Chuangxin, H., Yingzheng, L., and Kim, K. C. Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Physics of Fluids*, 31 (12):125111, 2019.

## A. Dataset Details

This section provides a detailed introduction to the large-scale high-fidelity flow field dataset, HFR-Bench, which was developed in-house. The dataset covers various grid types, including Cartesian grids, structured grids, and unstructured grids, to accommodate different fluid dynamics simulation needs. In terms of numerical simulation methods, HFR-Bench includes unsteady compressible Direct Numerical Simulations (DNS) and unsteady incompressible Reynolds-Averaged Navier-Stokes (RANS) methods, ensuring the capture of complex dynamic behaviors in the flow fields. Moreover, the dataset employs high-precision numerical techniques, with Cartesian and structured grids using the commonly applied Weighted Essentially Non-Oscillatory (WENO) scheme, while the unstructured grids are modeled using the Discontinuous Galerkin (DG) method to enhance the accuracy and reliability of the simulations. For each flow configuration, there are four grid resolutions and four levels of numerical precision, with high-fidelity and low-fidelity flow fields aligned based on real simulation time. Each flow field contains corresponding grid information and raw variables. With its diverse simulation settings, HFR-Bench provides rich data support for fluid dynamics research, advancing the development and validation of related algorithms and models.

### A.1. Cartesian Grid Flow Field Data

For the 2D flow fields with uniform Cartesian mesh, we choose the Riemann (RM), Rayleigh–Taylor instability (RT), and forward facing step (FFS) problems, which contain both discontinuities and complex flow features (Jing et al., 2003). The physical laws underlying our 2D flow fields are represented by the compressible Euler equations for the conservative variables, which can be expressed as

$$\mathbf{u}_t + f(\mathbf{u})_x + g(\mathbf{u})_y = \mathbf{0}, \quad (9)$$

where

$$\begin{aligned} \mathbf{u} &= (\rho, \rho u, \rho v, E)^T, \\ f(\mathbf{u}) &= (\rho u, \rho u^2 + p, \rho uv, u(E + p))^T, \\ g(\mathbf{u}) &= (\rho v, \rho u \bar{v}, \rho v^2 + p, v(E + p))^T. \end{aligned} \quad (10)$$

Here,  $\rho$  is the density,  $(u, v)$  is the velocity,  $E$  is the total energy,  $p$  is the pressure, and  $f(\mathbf{u})$  and  $g(\mathbf{u})$  are the convective fluxes. Four different high-order weighted essentially non-oscillatory (WENO) schemes are employed to solve these equations. For the conservation laws, the derivative  $f(\mathbf{u})_x$  at  $(x_i, y_j)$  is approximated along the line  $y = y_j$  by a convective flux difference:

$$f(\mathbf{u})_x|_{x=x_i} \approx \frac{1}{\Delta x} (\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}), \quad (11)$$

where, for the  $k$ th-order WENO scheme, the numerical flux  $\hat{f}_{i+\frac{1}{2}}$  depends on  $2k-1$  values  $f(\mathbf{u}_{mj})$ ,  $m = i-k-1, \dots, i+k-1$ . We set  $k$  to 3, 5, 7, or 9 to give third-order (WENO3), fifth-order (WENO5), seventh-order (WENO7), and ninth-order (WENO9) versions. Time discretization is achieved via the third-order Runge–Kutta method, and the Lax–Friedrichs building blocks and local characteristic decomposition are employed.

**Riemann (RM) Problem.** The two-dimensional RM problem containing multiple dimensions in the flow field is a common example of testing the efficiency of numerical methods. It solves the two-dimensional Euler equations in the unit square region  $[0, 1] \times [0, 1]$ , with initial conditions as follows:

$$(\rho, u, v, p) = \begin{cases} (1.5, 0, 0, 1.5), & \text{for } 0 \leq x < 0.8, 0 \leq y < 1 \\ (0.5323, 0, 1.206, 0.3), & \text{for } 0.8 \leq x < 1, 0 \leq y < 0.8 \\ (0.138, 1.206, 1.206, 0.029), & \text{for } 0.8 \leq x < 1, 0.8 \leq y < 1 \\ (0.532, 1.206, 0, 0.3), & \text{for } 0 \leq x < 0.8, 0.8 \leq y < 1 \end{cases} \quad (12)$$

The computational grid sizes used are  $200 \times 200$ ,  $400 \times 400$ ,  $800 \times 800$ , and  $1600 \times 1600$ , with the simulation time set to  $t = 0.8$ . The time step for storage is set to 0.0016. Figure 8(a) illustrate the flow field results generated by the WENO method under different grid resolutions. At the same grid resolution, the flow field generated by WENO methods with different accuracies exhibit distinct multi-scale flow field structures. Similarly, for the same accuracy of the WENO method, flow field cloud plots at different grid resolutions also display varying multi-scale flow field structures. Both numerical precision and grid resolution significantly enhance the multi-scale resolution.

**Rayleigh–Taylor Instability (RT) Problem.** In this problem, two fluids with different densities interact under gravity, with the denser fluid moving downward and the lighter fluid moving upward, which leads to the development of instability. The computational domain is defined as  $[0, 0.25] \times [0, 1]$  with initial conditions:

$$(\rho, u, v, p) = \begin{cases} (2, 0, -0.025\sqrt{\gamma p/\rho} \cos(8\pi x), 2y + 1), & 0 \leq y < 0.5 \\ (1, 0, -0.025\sqrt{\gamma p/\rho} \cos(8\pi x), y + 1.5), & 0.5 \leq y \leq 1 \end{cases} \quad (13)$$

where the heat parameter is specified as  $\gamma = 5/3$ . The boundary conditions for the left and right sides are taken as reflective, while the top and bottom boundaries are assigned with constant values:

$$(\rho, u, v, p)(x, y, t) = \begin{cases} (1, 0, 0, 2.5), & y = 1 \\ (2, 0, 0, 1.0), & y = 0 \end{cases} \quad (14)$$

The computation ends at  $t = 1.95$ , with a flow field storage timestep of 0.0039. The grid resolutions used are  $120 \times 480$ ,  $240 \times 960$ ,  $480 \times 1920$ , and  $960 \times 3840$ . Figure 8(b) shows the flow field results generated using different accuracy WENO methods at the same grid resolution. As observed, with increases in numerical precision and grid resolution, the vortex structures in the flow field become clearer and more intricate.

**Forward Facing Step (FFS) Problems.** This problem is commonly used in high-order numerical schemes and has been well-received in numerical testing, particularly for its ability to capture the instabilities in the fluid flow due to the presence of a source at the Mach stem and the resulting vortex formations, which are of great interest in fluid dynamics. The computational domain is defined as  $[0, 0.6] \times [0, 0.2] \cup [0, 3] \times [0.2, 1]$ , which corresponds to a small wind tunnel with a length of 3 and a height of 1, where a step with a length of 2.4 and a height of 0.2 is placed at a position 0.6 units away from the left inlet of the wind tunnel. The initial flow field in the computational region is given by:

$$(\rho, u, v, p) = (1.4, 3.0, 0, 1). \quad (15)$$

The simulation ends at  $t = 4$ , with a flow storage timestep of 0.008. The grid resolutions used are  $180 \times 48 \cup 36 \times 12$ ,  $360 \times 96 \cup 72 \times 24$ ,  $720 \times 192 \cup 144 \times 48$ , and  $1440 \times 384 \cup 288 \times 96$ . Figure 9(a) present the flow field results generated using different accuracy WENO methods at the same grid resolution. As can be observed, as the numerical precision and grid resolution increase, the vortex structures become clearer.

**Shock-longitudinal Vortex Interaction (SV) Problem.** For our 3D SV flow field, the underlying physical laws are represented by the following 3D unsteady Navier–Stokes equations for the conservative variables:

$$\begin{aligned} \mathbf{u}_t + f(\mathbf{u})_x + g(\mathbf{u})_y + h(\mathbf{u})_z &= \frac{1}{Re} (f_v(\mathbf{u})_x \\ &+ g_v(\mathbf{u})_y + h_v(\mathbf{u})_z), \end{aligned} \quad (16)$$

where  $Re$  is the Reynolds number and  $f_v(\mathbf{u})$ ,  $g_v(\mathbf{u})$ ,  $h_v(\mathbf{u})$  are the viscous fluxes. A stationary shock is initially located at the  $x = 0$  plane. Upstream of the shock ( $x \leq 0$ ),

$$\rho = 1, \quad u_x = -\gamma^{0.5} M_1, \quad u_r = u_\theta = 0, \quad p = 1, \quad (17)$$

where  $u_x$ ,  $u_r$ , and  $u_\theta$  are the axial, radial, and azimuthal velocity components. The downstream mean solution ( $x \geq 0$ ) is

$$\begin{aligned} \rho &= \frac{(\gamma + 1)M_1^2}{(\gamma - 1)M_1^2 + 2}, \quad u_x = -\frac{\gamma^{0.5}((\gamma - 1)M_1^2 + 2)}{(\gamma + 1)M_1^2}, \\ u_r = u_\theta &= 0, \quad p = \frac{2\gamma M_1^2 - (\gamma - 1)}{\gamma + 1}. \end{aligned} \quad (18)$$

An isentropic vortex is superimposed on the mean flow upstream of the shock. The axis of the vortex is along the  $x$ -axis ( $y = z = 0$ ). The perturbations in the azimuthal velocity and temperature associated with the vortex are expressed as

$$\begin{aligned} u'_\theta &= \frac{\varepsilon r}{2\pi} e^{0.5(1-r^2)}, \\ T' &= -\frac{(\gamma - 1)\varepsilon^2}{8\gamma\pi^2 r_0^2} e^{(1-r^2)}, \end{aligned} \quad (19)$$

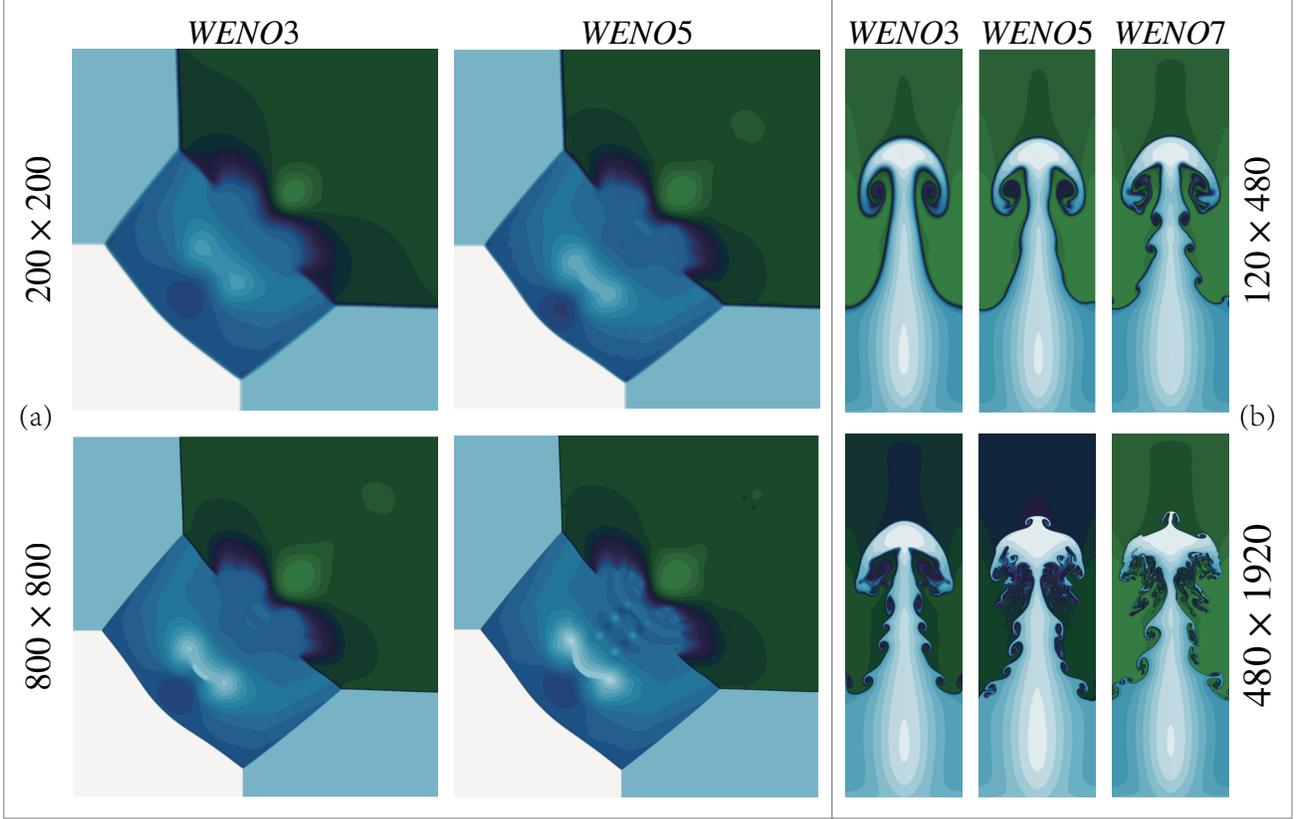


Figure 8. (a)  $x$ -direction velocity component  $u$  of dataset RM contours with 25 equally spaced isosurfaces in different grids and computation accuracy. (b)  $\rho$  flow field of RT in different grids and computation accuracy.

where  $r = \sqrt{y^2 + z^2}$  is the radius to the vortex axis,  $r_0$  is the vortex core radius, and  $\varepsilon$  is a dimensionless circulation at  $r = 1$ . In our simulations, the value of  $Re$ ,  $\gamma$ ,  $M_1$ , and  $\varepsilon$  are set to  $10^6$ , 1.4, 2, and 7, respectively. The grid sizes are  $96 \times 32 \times 32$ ,  $192 \times 64 \times 64$ ,  $384 \times 128 \times 128$ , and  $768 \times 256 \times 256$ . Moreover, the final simulation time is 11.0 s and the time step is 0.11 s. Therefore, 1,600 ( $1 \times 4 \times 4 \times 100$ ) 3D snapshots are separately collected. Figure 9(b) display the component  $u$  and  $v$  contours for SV for WENO3 and WENO5 using the  $96 \times 32 \times 32$  and  $384 \times 128 \times 128$  grids, respectively. We can clearly observe a significant difference between the high-fidelity results and their low-fidelity counterparts. The grid-resolution (GR) and numerical-precision (NP) of four datasets are listed in Table 1.

## A.2. Flow Field Data with Non-uniform meshes

For the non-uniform meshes, we select the circular jet standard test case. The maximum grid spacing difference and the highest precision can reach 64 times and 9 orders of magnitude, respectively. The control method is a two-dimensional incompressible RANS equation group. The Reynolds number  $re$  is 200, and the calculation time is stopped at  $t = 600.0$ , with a time step  $dt = 0.11$ . Figure 10 shows the flow field contours under different precision WENO grid models for the same grid model. As can be seen from the figures, as the precision and grid refinement increase, the flow field's turbulence becomes clearer.

## B. Experimental Details

Our model is implemented with Pytorch 1.10 (Paszke et al., 2019), and all experiments are conducted on a single NVIDIA A100 80GB GPU. The batch size (BS), memory (MM) and experimental variable and time per epoch are listed in Table 1. Our method and all the baseline methods are trained with the MSE loss in 2000 epochs. Learning ratio is set to  $1e - 5$  and decrease after 20 epochs if there is no loss degradation on the training set and the max epochs is 2000. (Every

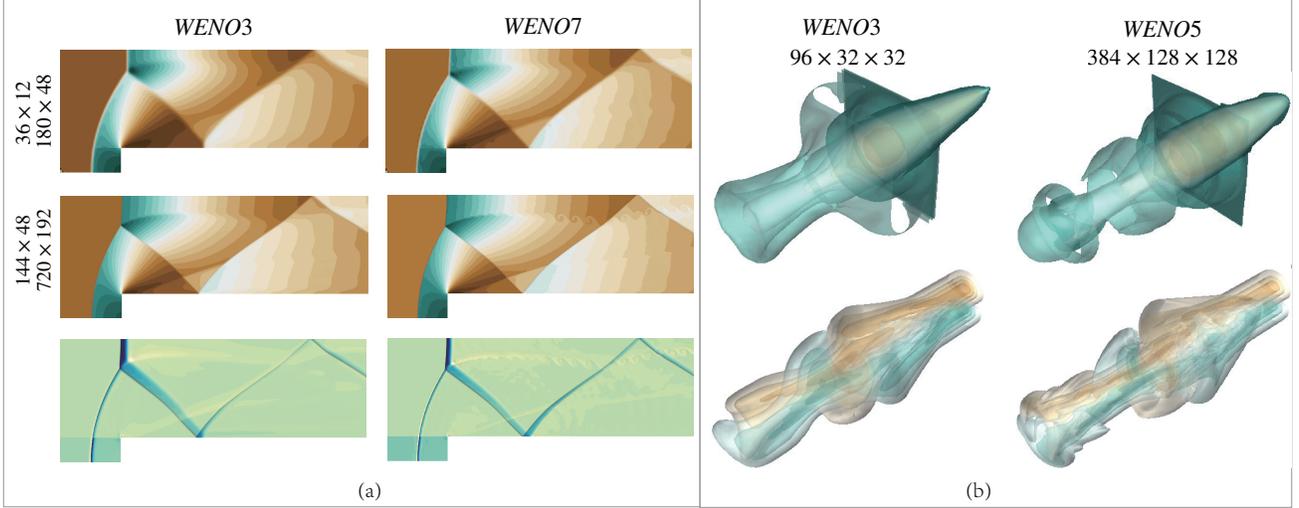


Figure 9. (a)  $x$ -direction velocity component  $u$  of dataset FFS contours with 25 equally spaced isosurfaces in different grids and computation accuracy and the bottom row displays the difference between low-resolution and high-resolution within the same computation scheme. (b) The two rows display  $x$ -direction velocity component  $u$  and  $y$ -direction velocity component  $v$  in the flow field of SV in different grids and computation accuracy.

method including ours can converge within 1000 epochs). During the training process, we adopt the AdamW optimizer for optimization. During design, we can set the number of residual layers of a Resu MLP to 10 and the max number of neurons in a Resu MLP is 64. To promote the learning of bases and ensure consistency of time series across different dimensions, we normalized the time series during training and performed inverse normalization when outputting the results. For the other models compared in the table, we utilized their original code and conducted experiments by only varying the input length.

PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), CORR (Correlation Coefficient) and DD (Dissipation difference) results are used to evaluate the quality of the synthesized  $HF$  flow fields. These three metrics are defined below, where the high-fidelity flow field and synthesized flow field are  $F^H$  and  $F^S$ , respectively.

**PSNR.** This represents the ratio of the maximum possible power of the signal and the destructive noise power that affects its representation accuracy. PSNR is computed using the mean squared error by the following formula:

$$PSNR(F^H, F^S) = 10 \times \log_{10} \left( \frac{(2^n - 1)^2}{MSE(F^H, F^S)} \right). \quad (20)$$

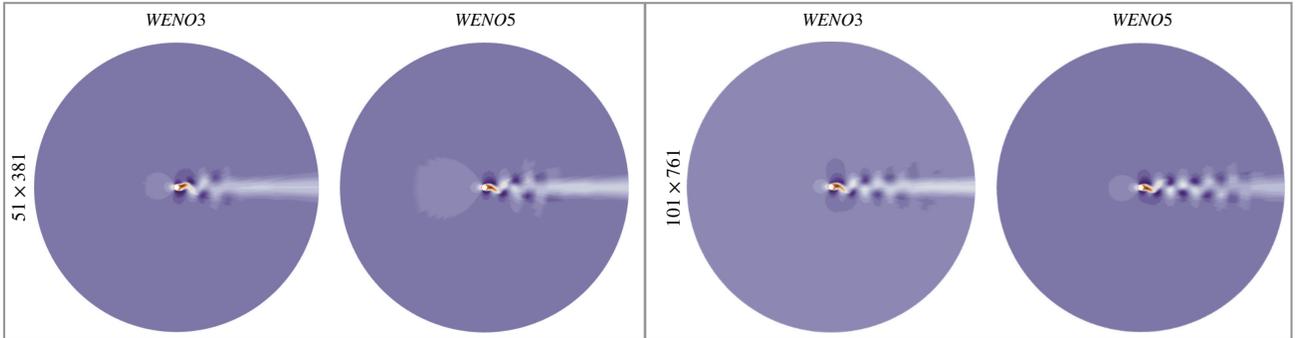


Figure 10.  $x$ -direction velocity component  $u$  of dataset cylinder contours with 25 equally spaced isosurfaces in different grids and computation accuracy.

**SSIM.** This is a measure of the similarity of two images, mimicking human perception by focusing primarily on edge and texture similarity. It is calculated by the following formula:

$$\text{SSIM}(F^H, F^S) = \frac{(2\mu_{F^H}\mu_{F^S} + c_1)(2\sigma_{F^H F^S} + c_2)}{(\mu_{F^H}^2 + \mu_{F^S}^2 + c_1)(\sigma_{F^H}^2 + \sigma_{F^S}^2 + c_2)}, \quad (21)$$

where  $\mu_{F^H}$  and  $\mu_{F^S}$  are the average values of  $F^H$  and  $F^S$ , respectively.  $\sigma_{F^H}^2$  and  $\sigma_{F^S}^2$  are the variance of  $F^H$  and  $F^S$ , respectively.  $\sigma_{F^H F^S}$  is the covariance of  $F^H$  and  $F^S$ .

**CORR.** The correlation coefficient reflects the degree of correlation between variables, and is defined as:

$$\text{CORR}(F^H, F^S) = \frac{\sum_{i=1}^n (F^S - \mu_{F^S})(F^H - \mu_{F^H})}{\sqrt{\sum_{i=1}^n (F^S - \mu_{F^S})^2} \sqrt{\sum_{i=1}^n (F^H - \mu_{F^H})^2}}. \quad (22)$$

**Dissipation difference.** The dissipation operator is used to assess the model performance in terms of capturing the discontinuities and rotating features. The dissipation of each velocity component ( $u$ ,  $v$ , and  $w$ ) is evaluated. The dissipation operator is defined by

$$\chi(F) = \left(\frac{\partial \mathbf{F}}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{F}}{\partial y}\right)^2 + \left(\frac{\partial \mathbf{F}}{\partial z}\right)^2. \quad (23)$$

The dissipation experiments aim to measure the difference in flow gradient between the ground truth data (GT) and the reconstructed data given by baselines and our method. This is represented by  $|\chi(Q^d) - \chi(\hat{Q}^d)|$ , with lower values indicating better performance.