# Semi-Supervised Learning for High-Fidelity Fluid Flow Reconstruction

**Cong Fu**
Texas A&M University
congfu@tamu.edu

**Jacob Helwig**
Texas A&M University
jacob.a.helwig@tamu.edu

**Shuiwang Ji**
Texas A&M University
sji@tamu.edu

## Abstract

Physical simulations of fluids are crucial for understanding fluid dynamics across many applications, such as weather prediction and engineering design. While high-resolution numerical simulations can provide substantial accuracy in analysis, it also results in prohibitive computational costs. Conversely, lower-resolution simulations are computationally less expensive but compromise the accuracy and reliability of results. In this work, we propose a cascaded fluid reconstruction framework to combine large amounts of low-resolution and limited amounts of paired high-resolution direct simulations for accurate fluid analysis. Our method can improve the accuracy of simulations while preserving the efficiency of low-resolution simulations. Our framework involves a proposal network, pre-trained with small amounts of high-resolution labels, to reconstruct an initial high-resolution flow field. The field is then refined in the frequency domain to become more physically plausible using our proposed refinement network, known as ModeFormer, which is implemented as a complex-valued transformer, with physics-informed unsupervised training. Our experimental results demonstrate the effectiveness of our approach in enhancing the overall performance of fluid flow reconstruction. The code will be made publicly available at https://github.com/divelab/AIRS/tree/main/OpenPDE/CFRF.

## 1 Introduction

Fluid dynamics has widespread applications in diverse fields of science and engineering, such as weather prediction, engineering design, and medical applications. It studies the behavior of fluids in motion, and its principles are used to understand and predict the behavior of fluids under different conditions. Direct Numerical Simulation (DNS) [1] is an important computational method to solve the Navier-Stokes equations, which are the fundamental equations governing fluid motions. Although DNS can resolve all the spatial and temporal scales of fluid motion, it requires fine-grained spatial and temporal discretization to achieve accurate simulation results for further fluid analysis, resulting in very high computational costs. The resolution needed rapidly grows when the Reynolds number increases. To reduce the computational cost, several other numerical methods are developed, such as Large Eddy Simulation (LES) [2] and Reynolds-Averaged Navier-Stokes (RANS) [3] methods, to only directly resolve the larger scales. These methods increase the simulation efficiency at the expense of low-fidelity simulation results.

Recently, machine learning has emerged as a promising approach for advancing scientific endeavors [4, 5], spanning diverse fields such as quantum mechanics [6, 7], biological science [5, 8, 9], material science [10, 11], and physical simulation [12]. One category of methods, aimed at accelerating physical simulation or solving partial differential equations with machine learning, involves learning a neural network surrogate model, also known as a neural solver [12–19]. This neural solver replaces traditional solvers and enables autoregressive prediction of solutions from one time step to the next. Another category of methods involves simulating low-resolution solutions using classical solvers and employing machine learning models to reconstruct high-fidelity fluid fields from the cheaply obtained low-resolution fields, thereby bridging the gap between computational feasibility and a

C. Fu et al., Semi-Supervised Learning for High-Fidelity Fluid Flow Reconstruction. *Proceedings of the Second Learning on Graphs Conference (LoG 2023)*, PMLR 231, Virtual Event, November 27–30, 2023.

realistic representation of fluid dynamics [20–22]. However, the majority of these methods rely heavily on a substantial amount of high-resolution labels to facilitate supervised learning [21, 23–25] and solutions may not be physically consistent. While some studies have explored unsupervised learning approaches for fluid field reconstruction, they either still require unpaired high-resolution fields [26] or are limited to handling simple laminar flow scenarios [27].

In this work, we present a systematic and realistic approach for solving fluid simulation problems. We begin by simulating a large amount of low-resolution solutions and then only a limited set of paired high-resolution solutions. To reconstruct the high-resolution fluid fields, we propose a cascaded fluid reconstruction framework consisting of a proposal network and a refinement network. The proposal network is trained in a supervised manner using the available high-resolution fluid field labels, enabling it to generate an initial high-resolution solution. Subsequently, the field undergoes refinement in the frequency domain to become more physically plausible using our refinement network trained with a physics-informed loss in an unsupervised manner. Our experimental results demonstrate the effectiveness of our approach in accurately reconstructing high-resolution fluid flow. Note that, in our context, the terms "resolution" and "fidelity" carry the same meaning and thus are used interchangeably throughout this paper.

## 2 Preliminaries

In Section 2.1, we begin by providing a definition of partial differential equations and then outline our proposed task for high-fidelity fluid flow reconstruction. In Section 2.2, we highlight the distinctions between high-fidelity fluid flow reconstruction and image super-resolution.

### 2.1 Problem Setup

**Partial Differential Equations.** The dynamics of a fluid flow system are governed by partial differential equations (PDEs). A PDE has the form

$$\frac{\partial \boldsymbol{u}}{\partial t} = F(\boldsymbol{x}, \boldsymbol{u}, \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}, \frac{\partial^2 \boldsymbol{u}}{\partial \boldsymbol{x}^2}, \cdots), \tag{1}$$

where $F$ is the function of independent variable $\boldsymbol{x} = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{1\times n}$, the state variable $\boldsymbol{u}(t, \boldsymbol{x}) = [u_1, u_2, \cdots, u_m] \in \mathbb{R}^{1\times m}$, and its partial derivatives with respect to $x_i$ up to a certain order. The objective is to solve the state $\boldsymbol{u}(t, \boldsymbol{x})$ given the initial condition $\boldsymbol{u}(0, \boldsymbol{x})$ and the boundary condition $\boldsymbol{u}(t, \boldsymbol{x} \in \partial\Omega)$, where $\Omega$ denotes the physical domain and $\partial\Omega$ denotes the domain boundary. Classical solvers for solving PDEs typically discretize the domain and approximate the derivatives using methods such as the finite difference method (FDM), finite volume method (FVM), or finite element method (FEM). Although solving PDEs with classical solvers like FEM can provide accurate predictions and error guarantees, it requires discretizing space and time at very fine resolutions, leading to slow and inefficient computations. On the other hand, low-resolution simulations offer greater efficiency but often produce less accurate solutions.

**High-Fidelity Fluid Flow Reconstruction.** To address the accuracy limitations of low-resolution simulations while retaining their efficiency, we propose a novel semi-supervised learning task that reconstructs high-resolution (HR) solutions from low-resolution (LR) solutions using a limited amount of high-resolution labels and a large amount of low-resolution simulation results. Specifically, we consider a fluid flow governing PDE $F$ and low-resolution and high-resolution domain discretizations $\Omega_L$ and $\Omega_H$, respectively.

To solve this PDE efficiently, we initially employ a classical solver $\mathcal{C}$ to simulate $k$ LR solutions

$$\{\boldsymbol{u}_L^{(I_i)}(t, \boldsymbol{x}) | \boldsymbol{u}_L := \mathcal{C}(F, \Omega_L), x \in \Omega_L, t \in [0, T], 0 \leq i \leq k\}, \tag{2}$$

where $I_i$ denotes different initial conditions. Subsequently, we generate a limited amount of HR solutions

$$\{\boldsymbol{u}_H^{(I_i)}(t, \boldsymbol{x}) | \boldsymbol{u}_H := \mathcal{C}(F, \Omega_H), x \in \Omega_H, t \in [0, T' \leq T], 0 \leq i \leq q < k\} \tag{3}$$

that are paired with a subset of $\boldsymbol{u}_L$. The objective is to reconstruct unlabeled LR solutions $\{\boldsymbol{u}_L / \boldsymbol{u}_H\}$ into HR solutions by using both information from $\boldsymbol{u}_L$ and $\boldsymbol{u}_H$. This task is both realistic and practically useful, as it allows for the preservation of computational efficiency in the initial simulation while enhancing the accuracy of the final solution through reconstruction.

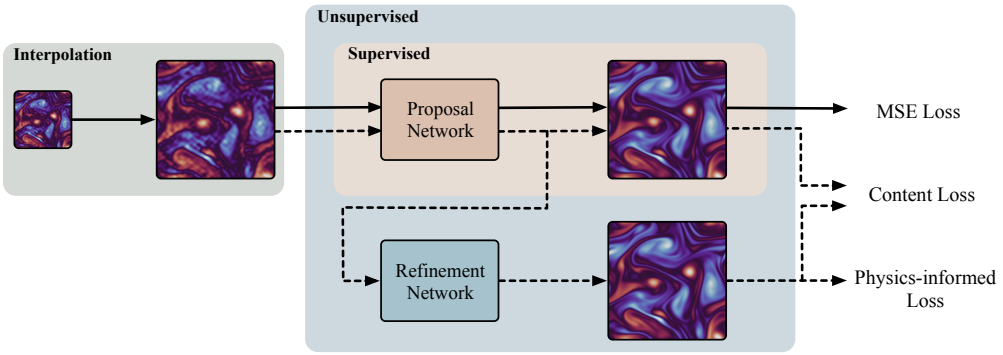


**Figure 1:** Illustration of the Cascaded Fluid Reconstruction Framework (CFRF). The input is interpolated to match the size of the high-resolution solutions. The CFRF comprises a proposal network and a refinement network. In the first stage, the proposal network is trained in a supervised manner, as indicated by the solid arrow branch. The second stage involves training the refinement network in an unsupervised manner, as represented by the dashed arrow branch. During inference, the input is sequentially processed by the proposal network and the refinement network to reconstruct the high-fidelity solution.

### 2.2 Difference between High-fidelity Fluid Flow Reconstruction and Image Super-resolution

We emphasize that high-fidelity fluid flow reconstruction is distinct from the image super-resolution task. In image super-resolution, the underlying assumption is that LR and HR images are observations of the same scene, with LR images typically obtained through the degradation of HR images. This relationship can be represented as:

$$I_x = \mathcal{D}(I_y; \sigma), \tag{4}$$

where $I_x$ are LR images and $I_y$ are HR images. $\mathcal{D}$ is the degradation function and $\sigma$ is the parameters of the degradation process, usually involving downsampling or the addition of Gaussian noise [28]. Consequently, the image super-resolution task aims to learn the reverse degradation process.

However, in high-fidelity fluid flow reconstruction, LR solutions are not directly derived from HR solutions. Instead, LR and HR solutions for fluid flows are obtained through numerical solvers using different domain discretizations. The coarse discretization of the LR simulation introduces inaccuracies and truncation errors. Moreover, the reconstructed HR solutions must adhere to the underlying partial differential equation, which imposes additional challenges on this task.

## 3 Methods

In this section, we begin by introducing our cascaded fluid reconstruction framework in Section 3.1. Following that, we delve into the motivation and architecture of our proposed refinement network, known as ModeFormer, in Section 3.2. Next, in Section 3.3, we outline the training strategy and the employed loss functions during the supervised and unsupervised training stages.

### 3.1 Cascaded Fluid Reconstruction Framework

In order to make the best use of limited HR labels and a large amount of unlabeled LR solutions, we propose a cascaded fluid reconstruction framework, which is illustrated in Figure 1. The framework consists of two stages: a supervised proposal stage and an unsupervised refinement stage. The objective of the supervised proposal stage is to produce an initial set of HR solutions that are as accurate and realistic as possible with the guidance of limited HR labels. Subsequently, the unsupervised refinement stage can adjust the initial reconstructions to further improve the reconstruction performance to make it more physically plausible. The second stage involves using a large amount of unlabeled LR solutions to train the refinement network. By doing so, the framework can leverage the vast amount of available data to improve the quality of the reconstructions. In the following, we will describe these two stages in detail and explain how they work together to achieve high-quality reconstructions.

**Supervised Proposal Network.** Given a limited amount of paired HR-LR solutions, it is natural to use a proposal network to learn an initial mapping between the two. However, due to the small data size of paired LR-HR solutions, we need to adopt a lightweight network for this stage. Therefore, we choose to use SRCNN [29] in this stage, which is specifically designed for image super-resolution tasks and consists of only three convolutional layers: patch extraction, non-linear mapping, and reconstruction layers. To implement SRCNN in our framework, we use kernel sizes of $9 \times 9$, $5 \times 5$, and $5 \times 5$ for these three layers, respectively. Since SRCNN uses the pre-upsampling scheme, we employ bicubic interpolation to upsample the LR solutions, which then serve as the input to the network.

**Unsupervised Refinement Network.** Once the proposal network has produced an initial reconstruction result, we can further improve the physical consistency of the reconstructed high-resolution field through the refinement network. These two networks are connected in a cascaded manner, which means the output of the proposal network is fed into the refinement network to generate the final result. To train the refinement network, we use a large amount of available unlabeled LR solutions in an unsupervised manner with physics-informed loss, which will be introduced in Section 3.3. In order to effectively and efficiently learn the data representation in this stage, we transfer the fluid flow into the frequency domain and predict the frequency modes using our proposed frequency-based transformer model, known as ModeFormer. The motivation and design details of the ModeFormer will be described in Section 3.2.

### 3.2 ModeFormer

For the refinement network, we propose a complex-valued transformer model to enhance the frequency modes of the output produced by the proposal network. We will first describe the motivation for learning in the frequency domain as opposed to the spatial domain. Then, we introduce the network structure of our proposed ModeFormer.

**Motivation for Learning in Frequency Domain.** In fluid dynamics, the frequency information carries specific meanings. Low-frequency modes in fluid flow capture slowly varying patterns, such as large vortices, effectively representing the overall behavior of the flow. Conversely, high-frequency modes correspond to rapidly varying patterns, like turbulence or small eddies, associated with localized and transient behavior in the flow.

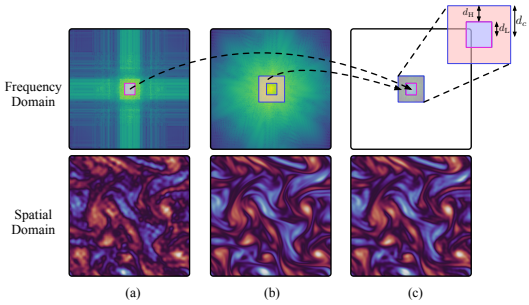


**Figure 2:** Demonstration of the motivation for learning in the frequency domain. (a) is the bicubic interpolation result and (b) is the HR label. (c) shows the result by combining low-frequency modes (area outlined with red square in (a)) from bicubic interpolation and high-frequency modes from the label (area outlined with blue ring in (b)). It turns out that (c) exhibits a similar fluid field to (b), which validates our motivation of only learning the high-frequency modes in a small ring area. $d_{\mathrm{L}}$ denotes the number of low-frequency modes to remain unchanged along $x$ and $y$ directions. $d_{\mathrm{H}}$ denotes the number of high-frequency modes to be replaced in $x$ and $y$ directions. Apart from the lowest $d_c$ ($d_c = d_{\mathrm{L}} + d_{\mathrm{H}}$) modes, the rest are filtered out.

In our experiments, we found two key points. First, low-frequency modes are relatively easier to recover, even with a simple bicubic interpolation. This allows the neural network to focus on recovering the high-frequency parts, which simplifies the problem and reduces computational complexity. Second, when recovering high-frequency modes, only a certain group of high-frequency modes are crucial, and the rest can be filtered out without a substantial loss of accuracy. For instance, as shown in Figure 2, (a) represents the bicubic interpolation of the low-resolution solution, and (b) represents the ground truth high-resolution solution. In (c), we retain the low-frequency modes (red square area) from the bicubic interpolation and retain a group of high-frequency modes (blue ring area) from the ground truth high-resolution solution and combine them together to form a solution. This approach is able to yield a resulting field that closely resembles the high-resolution ground truth.

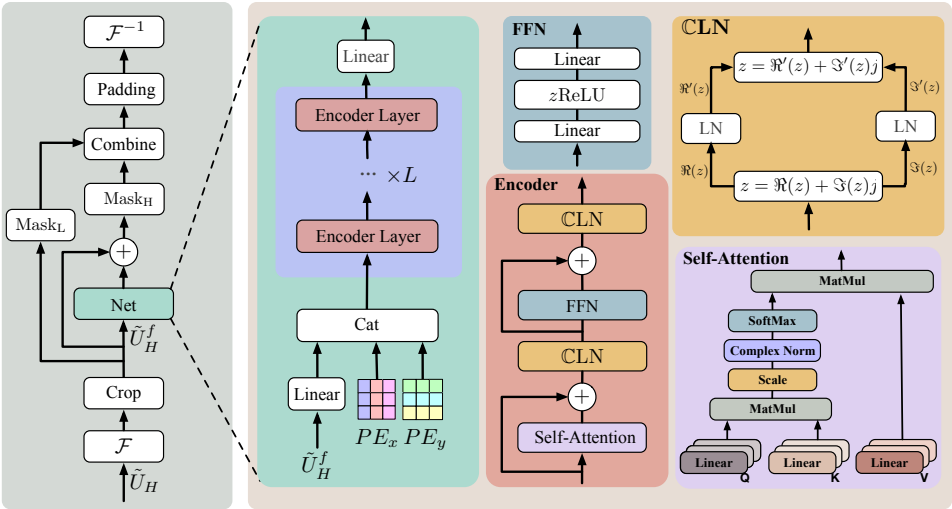


**Figure 3:** Architecture of the refinement network, ModeFormer. (a) shows the whole pipeline of the ModeFormer. (b) shows the detailed network design of the ModeFormer.

In practice, the number of frequency modes to be filtered out can be determined by progressively eliminating high-frequency modes from the ground truth high-resolution field and comparing the Euclidean distance between the filtered and original data until a noticeable discrepancy is observed. Also, we need to determine the number of low-frequency modes to be left unchanged from the bicubic interpolation. This is achieved by performing a similar process of high-frequency mode removal on the original data. However, instead of comparing the filtered result with the original ground truth, we compare it with the bicubic interpolation, which has the corresponding high-frequency parts removed. The stopping criterion is when the Euclidean distance between the two becomes sufficiently similar. The remaining frequency modes are the low-frequency ones that we aim to maintain unchanged.

**Network Architecture.** Let $\tilde{U}_H$ denote the output of the proposal network. In the ModeFormer, we begin by transforming $\tilde{U}_H$ into the frequency domain using the Fast Fourier Transform (FFT) algorithm. Subsequently, we extract only the central frequency modes such that

$$\tilde{U}_H^f = \text{Crop}^{d_c}(\mathcal{F}(\tilde{U}_H)), \tag{5}$$

where $\mathcal{F}$ denotes the Fourier transform and $\tilde{U}_H^f \in \mathbb{C}^{n_x \times n_y \times c}$ serves as the input to the subsequent blocks. $d_c$ denotes the number of frequency modes to retain, as shown in Figure 2.

For the positional encoding, since each mode has $x$ and $y$ coordinates, we calculate it separately along the $x$ and $y$ directions and then concatenate them to form the final positional encoding. Formally,

$$\text{PE}_{pos_x, pos_y, 2i} = \text{Cat}(\sin(pos_x/10000^{2i/d_{\text{PE}}}), \sin(pos_y/10000^{2i/d_{\text{PE}}})), \tag{6}$$

$$\text{PE}_{pos_x, pos_y, 2i+1} = \text{Cat}(\cos(pos_x/10000^{2i/d_{\text{PE}}}), \cos(pos_y/10000^{2i/d_{\text{PE}}})), \tag{7}$$

where $\text{Cat}(\cdot)$ is the concatenation and $d_{\text{PE}}$ is the size of positional encoding. Then, we employ a linear layer to project the input $\tilde{U}_H^f$ and concatenate it with the positional encoding such that

$$H = \text{Cat}(\text{Linear}(\tilde{U}_H^f), \text{PE}), \tag{8}$$

where $H \in \mathbb{C}^{n_x \times n_y \times d}$ serves as the input to the subsequent multi-head attention module [30]. We will flatten the first two dimensions of $H$, so the size of $H$ becomes $N \times d$, where $N$ equals $n_x \times n_y$. The multi-head attention module takes a query matrix $Q$, a key matrix $K$, and a value matrix $V$ as input. Formally, the attention mechanism is defined as follows:

$$Q_i = QW_i^Q, K_i = KW_i^K, V_i = VW_i^V, \text{ATT}_\text{i} = \text{softmax}(\frac{|Q_i K_i^T|}{\sqrt{d_k}})V_i, 1 \leq i \leq o \tag{9}$$

$$\text{MultiHead}(Q, K, V) = \text{Cat}(\text{ATT}_1, \text{ATT}_2, \cdots, \text{ATT}_\text{o})W^O, \tag{10}$$

where $o$ is the number of attention heads, $d_k$ denotes the embedding size of a single query or key vector, $\mathrm{Cat}(\cdot)$ is a concatenation operation, and $|\cdot|$ is the norm of a complex number. $W_i^Q \in \mathbb{C}^{d\times d_k}$, $W_i^K \in \mathbb{C}^{d\times d_k}$, $W_i^V \in \mathbb{C}^{d\times d_v}$, and $W_i^O \in \mathbb{C}^{hd_v\times d}$ are all learnable parameters with complexed values. Since we are using self-attention, the matrix $Q, K, V$ are the same so we simplify the notation of the multi-head attention layer as $\mathrm{MultiHead}(\cdot)$. The formal single transformer layer can be expressed as

$$\hat{H}^l = \mathbb{C}\mathrm{LN}(\mathrm{MultiHead}(H^l) + H^l), \tag{11}$$

$$H^{l+1} = \mathbb{C}\mathrm{LN}(\mathrm{FFN}(\hat{H}^l) + \hat{H}^l), \tag{12}$$

where $l$ denotes the $l$-th attention layer, $\mathbb{C}\mathrm{LN}$ is the complex layer normalization, and $\mathrm{FFN}$ represents the position-wise feed-forward network. Within the complex layer normalization, we apply separate layer normalization on both of the real and the imaginary part of the input. Formally,

$$\mathbb{C}\mathrm{LN} = \mathrm{LN}(\Re(z)) + j\mathrm{LN}(\Im(z)). \tag{13}$$

In the position-wise feed-forward network, we use a complex ReLU [31] that has the form

$$z\mathrm{ReLU}(z) = \begin{cases} z & \text{if } \theta_z \in (0, \pi/2], \\ 0 & \text{otherwise}, \end{cases} \tag{14}$$

where $\theta_z$ denotes the phase angle of the complex number $z$. After the final transformer layer, we use a linear projection layer to process the final embedding $H^L$, followed by adding a residual connection with the cropped input $\tilde{U}_H^f$. Next, we mask the low-frequency modes of the input $\tilde{U}_H^f$ and high-frequency modes of the output, and combine them together. Formally,

$$\hat{H}^L = {\mathrm{Mask}_\mathrm{H}}^{d_\mathrm{H}}(\mathrm{Linear}(H^L) + \tilde{U}_H^f) + {\mathrm{Mask}_\mathrm{L}}^{d_\mathrm{L}}(\tilde{U}_H^f), \tag{15}$$

where ${\mathrm{Mask}_\mathrm{L}}^{d_\mathrm{L}}(\cdot)$ and ${\mathrm{Mask}_\mathrm{H}}^{d_\mathrm{H}}(\cdot)$ represent the operations for masking the low-frequency modes and high-frequency modes, respectively. $d_\mathrm{L}$ denotes the number of low-frequency modes to retain from the input, and $d_\mathrm{H}$ represents the number of high-frequency modes to predict, as shown in Figure 2.

Finally, we pad the output frequency modes with zero values to match the size of the input. Then, we use inverse Fourier transform to convert the result back to the spatial domain, such that

$$\hat{U}_H = \mathcal{F}^{-1}(\mathrm{Pad}(\hat{H}^L)). \tag{16}$$

### 3.3 Training

We will first introduce loss functions used during training and then describe the training strategy. There are three loss functions in our method, namely MSE loss, physics-informed loss, and content loss.

**MSE Loss.** Mean squared error (MSE) loss is expressed as follows:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=0}^{N}(\hat{u_i} - u_i)^2, \tag{17}$$

where $\hat{\boldsymbol{u}}$ represents the reconstructions and $\boldsymbol{u}$ denotes the ground truth. $N$ is the number of grid points in the solution domain. The MSE loss quantifies the pixel-wise similarity between the reconstruction and the ground truth.

**Physics-informed Loss.** However, achieving a low MSE does not necessarily guarantee that the reconstructions adhere to the physical constraints imposed by the governing PDE. To ensure that the reconstructions satisfy the underlying PDE, we incorporate a physics-informed loss that penalizes inconsistencies with the fundamental physical laws. The idea of considering physics law in solving PDEs was originally proposed in PINN [32]. The physics-informed loss is formulated as follows:

$$\mathcal{R}(\hat{\boldsymbol{u}}) = \frac{\partial \hat{\boldsymbol{u}}}{\partial t} - F(\boldsymbol{x}, \hat{\boldsymbol{u}}, \frac{\partial \hat{\boldsymbol{u}}}{\partial \boldsymbol{x}}, \frac{\partial^2 \hat{\boldsymbol{u}}}{\partial \boldsymbol{x}^2}, \cdots), \tag{18}$$

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N}\sum_{i=0}^{N}(\mathcal{R}_i(\hat{\boldsymbol{u}}))^2, \tag{19}$$

where $\mathcal{R}(\hat{\boldsymbol{u}})$ represents the residual of the PDE. The right-hand side of Equation (18) corresponds to

the PDE being considered. If the predicted solution $\hat{\boldsymbol{u}}$ fails to satisfy the PDE, the residual will be non-zero. The physics-informed loss allows us to enforce adherence to the underlying physical laws during the reconstruction process.

**Content Loss.** While the physics-informed loss penalizes non-physical solutions, optimizing solely based on the physics-informed loss can lead to an ill-posed optimization problem. In some cases, solutions may satisfy the PDE but visually differ significantly from the ground truth. Therefore, we incorporate a content loss to ensure that the reconstructions maintain perceptual similarity with the reference solution by the proposal network. The concept of content loss was initially proposed by [33] for tasks such as style transfer and image super-resolution. It quantifies the semantic difference between two images by utilizing a pre-trained image classification network, such as the VGG network [34]. The form of content loss is

$$\mathcal{L}_{\text{content}} = \frac{1}{C_l H_l W_l} \left\| \phi_l(\hat{\boldsymbol{u}}) - \phi_l(\boldsymbol{u}) \right\|^2, \tag{20}$$

where $\phi$ denotes the pretrained network and $l$ denotes the $l$-th layer representation of the network $\phi$. The shape of the feature map after the $l$-th layer is denoted as $C_l \times H_l \times W_l$.

**Training Strategy.** For the first stage, we use paired LR-HR solutions to train the proposal network in a supervised way with MSE loss. Beyond MSE loss, we can also optionally add physics-informed loss in first-stage training. In Section 5.2, we compare the performance of models trained with these two variants of the loss function. For the second stage, we train the refinement network in an unsupervised manner with physics-informed loss and content loss. As described above, the content loss serves as a regularization term during the optimization of the refinement network. It ensures that the final reconstruction remains visually consistent with the output of the proposal network while simultaneously optimizing the physics-informed loss.

## 4 Related Works

In recent years, deep learning has emerged as a promising approach for accelerating computational fluid dynamics simulations and addressing the challenge of high-fidelity fluid flow reconstruction. Several notable advancements have been made in this field. For instance, Pant and Farimani [20] proposed a hybrid architecture combining MobileNet and UNet to reconstruct high-fidelity DNS data from filtered data. Yousif et al. [35] designed a multi-scale enhanced super-resolution generative adversarial network with a physics-based loss to reconstruct high-resolution laminar flows from spatially limited data. Fukami et al. [23, 24] developed a convolutional neural network leveraging skip-connection and multiscale convolutions for spatio-temporal fluid field reconstruction. Kim et al. [26] introduced a CycleGAN-based model to handle unpaired turbulence data for super-resolution reconstruction. Gao et al. [27] used CNN to reconstruct cardiovascular flow without requiring high-resolution labeling by considering the conservation law and boundary conditions. Ren et al. [21] designed a spatiotemporal super-resolution framework via physics-informed learning, employing convolutional-recurrent neural networks and post-upsampling strategy to refine temporal and spatial information. Esmaeilzadeh et al. [25] proposed MeshfreeFlowNet for reconstructing under-resolved solutions of Rayleigh–Benard convection equations, incorporating PDE constraints as an auxiliary loss. Shu et al. [22] introduced a physics-informed diffusion model to reconstruct fluid flow, relying solely on high-fidelity data during training. Several works such as Yang et al. [36] have proposed using attention for image super-resolution tasks, however, to our knowledge, ModeFormer represents the first application of the transformer architecture [30] to fluid flow reconstruction in the semi-supervised setting.

## 5 Experiments

We empirically demonstrate the effectiveness of our method for high-fidelity fluid flow reconstruction. In Section 5.1, we introduce the dataset, experimental setup, and benchmarking baseline models. In Section 5.2, we present the reconstruction performance of our method. To further validate the effectiveness of our proposed ModeFormer and assess the robustness of our framework to the amount of labeled data, we conduct several ablation studies in Section 5.3.

**Table 1:** Summary of the reconstruction performance for the 2D NS equation, where supervised models are trained with MSE loss and physics-informed loss. The size of labeled data is $1/16$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | RCAN | SRCNN | CFRF |
|---|---|---|---|---|---|---|---|
| MSE | 0.2286 | 0.1835 | 0.1175 | 0.0961 | 1.6193 | 0.0948 | **0.0946** |
| Equation Loss | 48.9912 | 1.4797 | 1.6489 | 0.9637 | 53.6201 | 0.2346 | **0.0976** |

**Table 2:** Summary of the reconstruction performance for the 2D NS equation, where supervised models are trained only with MSE loss. The size of labeled data is $1/16$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | RCAN | SRCNN | CFRF |
|---|---|---|---|---|---|---|---|
| MSE | 0.2286 | 0.1835 | 0.0977 | 0.0906 | 0.0862 | 0.0768 | **0.0748** |
| Equation Loss | 48.9912 | 1.4797 | 2.367 | 2.2075 | 5.7242 | 0.8847 | **0.2208** |

### 5.1 Experimental Setting

**Dataset.** We evaluate our method on 2D incompressible Navier-Stokes equations [37], 2D shallow water equations [38], and 2D diffusion-reaction equations [38]. The 2D incompressible Navier-Stokes equations are a commonly used PDE in the field of computational fluid dynamics and can be used to model various dynamics systems, such as turbulent dynamics and hydromechanical systems. The 2D shallow water equations are used for modeling free-surface flow problems. The 2D diffusion-reaction equation can be used to model biological pattern formation. The detailed equations for these three PDEs can be found in Appendix C.

**Setup.** For 2D NS equations, we simulate 20 trajectories of low-resolution solutions on a grid of size $64 \times 64$. The initial conditions for each trajectory are randomly initialized. The solutions are recorded at a time interval of $0.02s$. Out of the 20 trajectories, we split the train/valid/test as 80%/10%/10%. $1/16$ training trajectories have high-resolution solutions to serve as training labels. These high-resolution labels are simulated with a resolution of $1024 \times 1024$ and then downsampled to $256 \times 256$. To ensure consistency, the initial conditions of the low-resolution data are downsampled from their corresponding high-resolution counterparts. The simulation setting for 2D shallow water equations and 2D diffusion-reaction equations can be found in Appendix C. We assess the reconstruction performance using two quantitative metrics: (i) **MSE** evaluates the pixel-wise similarity between reconstructed fluid flow and the ground truth high-resolution flows; (ii) **Equation Loss** measures the extent to which the reconstructed solutions adhere to the governing physical laws. The equation loss calculation is identical to the physics-informed loss, as described in Section 3.3. Network hyperparameters and training details can be found in Appendix A and Appendix D, respectively.

**Baselines.** We use several numerical and deep learning methods as baselines. For the numerical methods, we consider the interpolation-based methods, including nearest-neighbor and bicubic interpolation. For the deep learning methods, we consider Fourier Neural Operator (FNO) [37], Super-Resolution Convolutional Neural Network (SRCNN) [29], Residual Dense Network (RDN) [39], and Residual Channel Attention Network (RCAN) [40]. More details about baselines can be found in Appendix B.

### 5.2 Results

**2D Incompressible Navier-Stokes Equations.** As discussed in Section 3.3, the proposal network can be trained using either the MSE loss alone or a combination of MSE loss and physics-informed loss. These options are also applicable to other supervised baselines. We conducted experiments using both training loss choices and summarize the results in Table 1 and Table 2, respectively. In our cascaded fluid reconstruction framework (CFRF), we employed SRCNN as the proposal network and ModeFormer as the refinement network. The results demonstrate that for both choices of supervised training loss, CFRF achieves the best performance in terms of MSE and equation loss. Moreover, this result shows that with the unsupervised refinement network, the physical consistency can be further enhanced upon the supervised proposal network. This indicates that the proposed framework effectively reconstructs high-fidelity fluid flow and highlights its ability to enforce the underlying physical constraints during the reconstruction process. More results and visualizations can be found in Appendix F and Appendix I, respectively. Regarding the SRCNN

**Table 3:** Comparison of the reconstruction performance of different refinement networks for the 2D NS equation. $\text{SRCNN}^{\text{P}}$ denotes SRCNN without refinement, while $\text{SRCNN}^{\text{R}}$ denotes $\text{SRCNN}^{\text{P}}$ followed by a separate SRCNN refinement network. The size of labeled data is $1/16$ of the whole training data. The proposal network is trained with MSE loss.

| Method | Nearest | Bicubic | $\text{SRCNN}^{\text{P}}$ | $\text{SRCNN}^{\text{R}}$ | $\text{ModeFormer}^{\text{R}}$ | $\text{UNet}^{\text{R}}$ |
|---|---|---|---|---|---|---|
| MSE | 0.2286 | 0.1835 | 0.0768 | 0.0795 | **0.0748** | 0.0762 |
| Equation Loss | 48.9912 | 1.4797 | 0.8847 | 0.4279 | **0.2208** | 0.8031 |

**Table 4:** Results of the reconstruction with different fractions of labeled data to train the proposal network for the 2D NS equation. $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{64}$, and $\frac{1}{128}$ denote the fraction of labeled data in the whole training data. † means the model is trained fully unsupervised without using proposal network. The proposal network is trained with MSE loss and physics-informed loss.

| Method | $\text{CFRF}^{\dagger}$ | $\text{CFRF}^{\frac{1}{8}}$ | $\text{CFRF}^{\frac{1}{16}}$ | $\text{CFRF}^{\frac{1}{32}}$ | $\text{CFRF}^{\frac{1}{64}}$ | $\text{CFRF}^{\frac{1}{128}}$ |
|---|---|---|---|---|---|---|
| MSE | 0.1738 | **0.0904** | 0.0946 | 0.0964 | 0.0954 | 0.0975 |
| Equation Loss | 0.3696 | 0.1017 | **0.0976** | 0.1054 | 0.1167 | 0.1198 |

baseline, incorporating the physics-informed loss results in a lower equation loss but slightly worse MSE. Since our CFRF employs SRCNN as the proposal network, the performance of CFRF is similarly influenced. Performance for FNO, RDN, and RCAN is worse than that of SRCNN. This discrepancy may arise due to the limited size of our labeled dataset. With a small amount of labeled data, the increased complexity of these models can lead to overfitting and thus result in poorer performance compared to SRCNN.

**2D Shallow Water Equations and 2D Diffusion-Reaction Equations.** For the shallow water equations, we test $2\times$ and $4\times$ reconstruction settings, where the resolution of the HR data is $2\times$ and $4\times$ that of the LR data, both with $1/20$ labeled training data. For the 2D diffusion-reaction equations, we test $8\times$ reconstruction setting, and $1/20$ training data have HR labels. We train the proposal network with only MSE loss for these two equations. The results can be found in Appendix E, which shows that our method can achieve the best MSE and equation loss. Visualizations can be found in Appendix I.

### 5.3 Ablation Studies

In this section, we conduct more experiments on the 2D NS equation to show the effectiveness of our proposed ModeFormer and the robustness of our framework to the amount of labeled data. First, we conduct ablation studies on the choices of the refinement network. We compared SRCNN, ModeFormer, and UNet as the refinement network options. Note that the SRCNN refinement network shares the same structure as the SRCNN proposal network, with weights initialized from the trained SRCNN proposal network. The results, shown in Table 3, demonstrate that our ModeFormer outperforms the other options, achieving the best results after the unsupervised training stage. The second ablation study focused on the sensitivity of the CFRF to the amount of labeled training data. As shown in Table 4, using labeled data leads to improved results compared to fully unsupervised learning. However, even when the amount of labeled data is reduced to only $1/64$ or $1/128$ of the whole training data, the MSE and equation loss only slightly deteriorate while still remaining significantly better than fully unsupervised learning.

## 6 Conclusions

In this work, we propose a cascaded fluid reconstruction framework to perform high-fidelity fluid field reconstruction with a large amount of unlabeled low-resolution solutions and a limited amount of paired high-resolution solutions. We first use a proposal network trained on labeled data to produce an initial reconstructed high-resolution flow field. Then, the field is refined by a refinement network which is trained using physics-informed unsupervised learning. The experimental results showcase that our method can achieve accurate and physically plausible reconstructions while maintaining the efficiency of low-resolution simulation.

## Acknowledgments

This work was supported in part by National Science Foundation grant IIS-2006861.

## References

[1] Parviz Moin and Krishnan Mahesh. Direct numerical simulation: a tool in turbulence research. *Annual review of fluid mechanics*, 30(1):539–578, 1998.

[2] J Janicka and AJPotCI Sadiki. Large eddy simulation of turbulent combustion systems. *Proceedings of the Combustion Institute*, 30(1):537–547, 2005.

[3] Brian Edward Launder and Bahrat I Sharma. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in heat and mass transfer*, 1 (2):131–137, 1974.

[4] Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, Keir Adams, Maurice Weiler, Xiner Li, Tianfan Fu, Yucheng Wang, Haiyang Yu, YuQing Xie, Xiang Fu, Alex Strasser, Shenglong Xu, Yi Liu, Yuanqi Du, Alexandra Saxton, Hongyi Ling, Hannah Lawrence, Hannes Stärk, Shurui Gui, Carl Edwards, Nicholas Gao, Adriana Ladera, Tailin Wu, Elyssa F. Hofgard, Aria Mansouri Tehrani, Rui Wang, Ameya Daigavane, Montgomery Bohde, Jerry Kurtin, Qian Huang, Tuong Phung, Minkai Xu, Chaitanya K. Joshi, Simon V. Mathis, Kamyar Azizzadenesheli, Ada Fang, Alán Aspuru-Guzik, Erik Bekkers, Michael Bronstein, Marinka Zitnik, Anima Anandkumar, Stefano Ermon, Pietro Liò, Rose Yu, Stephan Günnemann, Jure Leskovec, Heng Ji, Jimeng Sun, Regina Barzilay, Tommi Jaakkola, Connor W. Coley, Xiaoning Qian, Xiaofeng Qian, Tess Smidt, and Shuiwang Ji. Artificial intelligence for science in quantum, atomistic, and continuum systems, 2023.

[5] Meng Liu, Youzhi Luo, Limei Wang, Yaochen Xie, Hao Yuan, Shurui Gui, Haiyang Yu, Zhao Xu, Jingtun Zhang, Yi Liu, et al. Dig: A turnkey library for diving into graph deep learning research. *The Journal of Machine Learning Research*, 22(1):10873–10881, 2021.

[6] Dmitrii Kochkov, Tobias Pfaff, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Bryan K Clark. Learning ground states of quantum hamiltonians with graph networks. *arXiv preprint arXiv:2110.06390*, 2021.

[7] Cong Fu, Xuan Zhang, Huixin Zhang, Hongyi Ling, Shenglong Xu, and Shuiwang Ji. Lattice convolutional networks for learning ground states of quantum many-body systems. *arXiv preprint arXiv:2206.07370*, 2022.

[8] Meng Liu, Cong Fu, Xuan Zhang, Limei Wang, Yaochen Xie, Hao Yuan, Youzhi Luo, Zhao Xu, Shenglong Xu, and Shuiwang Ji. Fast quantum property prediction via deeper 2d and 3d graph networks. *arXiv preprint arXiv:2106.08551*, 2021.

[9] Cong Fu, Keqiang Yan, Limei Wang, Wing Yee Au, Michael McThrow, Tao Komikado, Koji Maruhashi, Kanji Uchino, Xiaoning Qian, and Shuiwang Ji. A latent diffusion model for protein structure generation. *arXiv preprint arXiv:2305.04120*, 2023.

[10] Keqiang Yan, Yi Liu, Yuchao Lin, and Shuiwang Ji. Periodic graph transformers for crystal material property prediction. In *The 36th Annual Conference on Neural Information Processing Systems*, 2022.

[11] Yuchao Lin, Keqiang Yan, Youzhi Luo, Yi Liu, Xiaoning Qian, and Shuiwang Ji. Efficient approximations of complete interatomic potentials for crystal property prediction. URL https://openreview.net/forum?id=KaKXygtEGK.

[12] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR, 2020.

[13] Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*, 2020.

[14] Zijie Li and Amir Barati Farimani. Graph neural network-accelerated lagrangian fluid simulation. *Computers & Graphics*, 103:201–211, 2022.

[15] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

[16] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

[17] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[18] Xiongye Xiao, Defu Cao, Ruochen Yang, Gaurav Gupta, Gengshuo Liu, Chenzhong Yin, Radu Balan, and Paul Bogdan. Coupled multiwavelet neural operator learning for coupled partial differential equations. *arXiv preprint arXiv:2303.02304*, 2023.

[19] Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant fourier neural operators for partial differential equations. *arXiv preprint arXiv:2306.05697*, 2023.

[20] Pranshu Pant and Amir Barati Farimani. Deep learning for efficient reconstruction of high-resolution turbulent dns data. *arXiv preprint arXiv:2010.11348*, 2020.

[21] Pu Ren, Chengping Rao, Yang Liu, Zihan Ma, Qi Wang, Jian-Xun Wang, and Hao Sun. Physics-informed deep super-resolution for spatiotemporal data. *arXiv preprint arXiv:2208.01462*, 2022.

[22] Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023.

[23] Kai Fukami, Koji Fukagata, and Kunihiko Taira. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019.

[24] Kai Fukami, Koji Fukagata, and Kunihiko Taira. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *Journal of Fluid Mechanics*, 909:A9, 2021.

[25] Soheil Esmaeilzadeh, Kamyar Azizzadenesheli, Karthik Kashinath, Mustafa Mustafa, Hamdi A Tchelepi, Philip Marcus, Mr Prabhat, Anima Anandkumar, et al. Meshfreeflownet: A physics-constrained deep continuous space-time super-resolution framework. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE, 2020.

[26] Hyojin Kim, Junhyuk Kim, Sungjin Won, and Changhoon Lee. Unsupervised deep learning for super-resolution reconstruction of turbulence. *Journal of Fluid Mechanics*, 910:A29, 2021.

[27] Han Gao, Luning Sun, and Jian-Xun Wang. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, 33(7):073603, 2021.

[28] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020.

[29] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[31] Nitzan Guberman. On complex valued convolutional neural networks. *arXiv preprint arXiv:1602.09046*, 2016.

[32] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[33] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Mustafa Z Yousif, Linqi Yu, and Hee-Chang Lim. High-fidelity reconstruction of turbulent flow from spatially limited data using enhanced super-resolution generative adversarial network. *Physics of Fluids*, 33(12):125119, 2021.

[36] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5791–5800, 2020.

[37] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*.

[38] Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

[39] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[40] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

# A Network Details

## A.1 Proposal Network

The proposal network employed in our cascaded fluid reconstruction framework (CFRF) is the Super-Resolution Convolutional Neural Network (SRCNN), as described in Section 3.1. In SRCNN, we use $9 \times 9$, $5 \times 5$, and $5 \times 5$ kernel size for patch extraction, non-linear mapping, and reconstruction layers, respectively. The channel numbers for the middle two convolutional layers are $64$ and $32$, respectively.

## A.2 Refinement Network

The refinement network employed in the CFRF is our proposed ModeFormer. By default, the input time step size or sliding window size is 7, i.e. the number of input channels is 7. After the linear projection of $\tilde{U}_H^f$, the embedding size becomes $24$. Positional encoding size $d_{\mathrm{PE}}$ for $x$ and $y$ coordinates are both $4$. The number of attention heads $o$ is 2, and the embedding size $d_k$ and $d$ are $32$. Within the position-wise feed-forward network, the hidden size is increased to $64$ and then decreased to $32$. The number of lowest frequency modes $d_L$ to remain unchanged from the input is $8$. The number of high-frequency modes $d_H$ to predict is $20$. For the shallow water equation, the sliding window size is 3, $d_L$ and $d_H$ are 5 and 23, respectively. For the diffusion-reaction equation, the sliding window size is $4$, $d_L$ and $d_H$ are $8$ and $28$, respectively. The embedding size $d_k$ and $d$ are $64$.

# B Baseline Methods Details

FNO is a resolution-invariant neural operator for solving partial differential equations. SRCNN is a lightweight deep learning model for image super-resolution tasks. Note that SRCNN also serves as the proposal network in our framework, as described in Section 3.1. RDN and RCAN are more advanced super-resolution networks. RDN uses residual connections and densely connected blocks to capture and propagate image features across multiple layers effectively. RCAN proposes a residual in residual (RIR) structure to form a very deep network to learn high-frequency information.

For the SRCNN baseline, the network hyperparameters are the same as the proposal network in Appendix A.1. For the RDN, the number of residual dense blocks is $4$, and the number of layers in each residual dense block is $3$. In each residual dense block, the growth rate is $32$. For the FNO, the number of Fourier layers is $4$, and the embedding size is $20$. Input time step size or sliding window size is 3 for training with MSE loss only and 7 for training with MSE plus physics-informed loss. For the RCAN, we use the default hyperparameters in their official code. The number of residual blocks is 16 and the number of residual groups is 10. The hidden channel number is 64 and the number of feature maps reduction is 16.

# C Data Details

**2D Incompressible Navier-Stokes Equation.** The 2D incompressible NS equation is expressed as follows:

$$\partial_t w(\boldsymbol{x}, t) + u(\boldsymbol{x}, t) \cdot \nabla w(\boldsymbol{x}, t) = \frac{1}{Re} \Delta w(\boldsymbol{x}, t) + f(\boldsymbol{x}), \tag{21}$$

$$\nabla \cdot u(\boldsymbol{x}, t) = 0, \tag{22}$$

$$w(\boldsymbol{x}, 0) = w_0(\boldsymbol{x}), \tag{23}$$

where $w(\boldsymbol{x}, t) \in \mathbb{R}$ is the vorticity field and $u(\boldsymbol{x}, t) \in \mathbb{R}^2$ is the velocity. $Re$ denotes the Reynolds number that controls the viscosity of the fluid. $w_0(\boldsymbol{x}) \in \mathbb{R}$ is the initial vorticity. The solution domain we consider is $x \in [0, 2\pi]^2, t \in [0, 10]$. $f(\boldsymbol{x})$ is the forcing term that represents the external forces exerting on the fluid flow. The forcing term we adopt is $f(x_1, x_2) = 0.1(\sin(2\pi(x_1 + x_2)) + \cos(2\pi(x_1 + x_2)))$. The initial condition $w_0(\boldsymbol{x})$ is sampled from a Gaussian random field $\mathcal{N} \sim (0, 7^{3/2}(-\Delta + 49I)^{-2.5})$ with periodic boundary conditions. The equation is solved by a pseudospectral method implemented by Li et al. [37]. As mentioned in Section 5.1, we simulate 20 low-resolution trajectories and use 80% (16) trajectories as training data. In our main experiment, for

the first training trajectory, we additionally simulate high-resolution solutions to serve as training labels, corresponding to $1/16$ training data. In Section 5.3, we perform an ablation study on the ratio of labeled training data, where we reduce the ratio to $1/32$, $1/64$, and $1/128$. In these cases, the amount of labeled data is less than one trajectory. To mimic a realistic scenario, we only run the high-resolution simulation from the initial time step to generate labeled data. Specifically, ratio $1/32$, $1/64$, and $1/128$ corresponds to time step $0$ to $250$, $0$ to $125$, and $0$ to $64$, respectively.

**2D Shallow Water Equation.** We generate the shallow water equation (SWE) data using scripts provided by PDEBench [38]. Shallow water equations are used for modeling free-surface flow problem and the form is shown below:

$$\partial_t h + \nabla \cdot (h\boldsymbol{u}) = 0, \tag{24}$$

$$\partial_t (h\boldsymbol{u}) + \nabla \cdot (h\boldsymbol{u}\boldsymbol{u}^T) + \frac{1}{2} g \nabla h^2 = 0, \tag{25}$$

where $\boldsymbol{u} = (u, v)$ denotes the horizontal and vertical direction velocities, $h$ denotes the water depth, and $g$ denotes the gravitational acceleration. The time $t$ is restricted to $[0, 1]$, and we consider a spatial domain $\Omega \in [-2.5, 2.5]^2$ with Neumann boundary conditions. Water height is initialized as a circular bump in the center of the domain, as shown below:

$$h(x, 0) = \begin{cases} 2.0 & r < \sqrt{x_1^2 + x_2^2} \\ 1.0 & r \geq \sqrt{x_1^2 + x_2^2} \end{cases}, \tag{26}$$

where the radius $r$ is randomly sampled from $\mathcal{U}(0.3, 0.7)$.

We simulate 100 trajectories and record 100 time steps for each trajectory. We split the train/valid/test as 80%/10%/10% and $1/20$ training data (corresponding to 4 trajectories) have HR labels and are used for training the proposal network. We test two resolution cases, $64 \times 64 \rightarrow 128 \times 128$ (2×) and $64 \times 64 \rightarrow 256 \times 256$ (4×).

**2D Diffusion-Reaction Equation.** We generate the 2D diffusion-reaction equation data using scripts provided by PDEBench [38]. The 2D diffusion-reaction equation is shown as below. This equation can be used to model biological pattern formation. In the equation, there are two non-linearly coupled variables called the activator $u$ and inhibitor $v$.

$$\partial_t u = D_u \partial_{xx} u + D_u \partial_{yy} u + R_u, \tag{27}$$

$$\partial_t v = D_v \partial_{xx} v + D_v \partial_{yy} v + R_v, \tag{28}$$

$D_u$ and $D_v$ are the diffusion coefficient for the activator and inhibitor, respectively. $R_u(u, v)$ and $R_v(u, v)$ are the activator and inhibitor reaction functions, as shown below.

$$R_u(u, v) = u - u^3 - k - v, \tag{29}$$

$$R_v(u, v) = u - v. \tag{30}$$

Parameters are $k = 5 \times 10^{-3}, D_u = 1 \times 10^{-3}, D_v = 5 \times 10^{-3}$, $t \in [0, 5]$, and the domain $\Omega \in (-1, 1)^2$ has Neumann boundary conditions. Initial conditions are sampled from Gaussian: $u(0, x, y) \sim \mathcal{N}(0, 1.0)$.

As with SWE, we simulate 100 trajectories and record 100 time steps for each trajectory[1]. We split the train/valid/test as 80%/10%/10% and $1/20$ training data (corresponding to 4 trajectories) have HR labels and are used for training the proposal network. We test resolution $16 \times 16 \rightarrow 128 \times 128$ (8×).

# D Training Details

To train the proposal network and other supervised baselines, we split the labeled data (e.g., $1/16$ of the whole training data for 2D NS equation) into a training and validation set with an 80%/20% ratio.

[1]Note that for the 2D diffusion-reaction equation, the LR solutions are obtained by downsampling the HR solutions. Since the HR initial condition is purely Gaussian noise, so downsampled LR initial condition is completely different from the original initial condition, which results in non-relevant trajectories if using LR simulation.

The best checkpoint is selected based on the lowest MSE obtained on the labeled validation data. For training the refinement network, all the training trajectories, including the labeled low-resolution data, are utilized. The best checkpoint is selected based on the lowest physics-informed loss achieved on the original validation data. During refinement network training, the parameters of the proposal network remain frozen. The weights assigned to the MSE loss, physics-informed loss, and content loss are uniformly set to 1. Finally, the performance of all models is evaluated on the hold-out test data.

We implement our method using PyTorch [41]. We adopt the Adam optimizer [42] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay $2e^{-4}$. We use the batch size of 16 for both the proposal network and the refinement network. The proposal network is trained for 200 epochs, while the refinement network is trained for 100 epochs. All models are trained on a single NVIDIA A6000 GPU.

# E Results for 2D Shallow Water Equations and 2D Diffusion-Reaction Equations

**Table 5:** Summary of the reconstruction performance for 2D shallow water equations on resolution $64$ to $128$, where supervised models are trained only with MSE loss. The size of labeled data is $1/20$ of the whole training data. The result of RCAN is excluded as the training is not stable.

| Method | Nearest | Bicubic | FNO | RDN | SRCNN | CFRF |
|---|---|---|---|---|---|---|
| MSE | $15.985e^{-5}$ | $9.479e^{-5}$ | $9.484e^{-5}$ | $21.950e^{-5}$ | $8.896e^{-5}$ | $\mathbf{8.797e^{-5}}$ |
| Equation Loss | $1.7152$ | $0.0715$ | $0.0715$ | $0.5414$ | $0.0768$ | $\mathbf{0.0543}$ |

**Table 6:** Summary of the reconstruction performance for 2D shallow water equation on resolution $64$ to $256$, where supervised models are trained only with MSE loss. The size of labeled data is $1/20$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | RCAN | SRCNN | CFRF |
|---|---|---|---|---|---|---|---|
| MSE | $2.969e^{-4}$ | $2.109e^{-4}$ | $2.110e^{-4}$ | $3.709e^{-4}$ | $7.815e^{-4}$ | $1.933e^{-4}$ | $\mathbf{1.916e^{-4}}$ |
| Equation Loss | $3.3401$ | $0.0583$ | $0.0585$ | $1.1761$ | $23.8961$ | $0.0747$ | $\mathbf{0.0451}$ |

**Table 7:** Summary of the reconstruction performance for 2D diffusion-reaction equation on resolution $16$ to $128$, where supervised models are trained only with MSE loss. The size of labeled data is $1/20$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | RCAN | SRCNN | CFRF |
|---|---|---|---|---|---|---|---|
| MSE | $5.6028e^{-2}$ | $4.5758e^{-2}$ | $3.3112e^{-2}$ | $2.0481e^{-2}$ | $2.1836e^{-2}$ | $2.0398e^{-2}$ | $\mathbf{2.0396e^{-2}}$ |
| Equation Loss | $12.5671$ | $2.8352$ | $2.5241$ | $0.5369$ | $0.8093$ | $1.5681$ | $\mathbf{0.2137}$ |

# F More Results for the 2D NS Equation

We present additional comparison results with baseline models in Table 8 and Table 9, where supervised models are trained using MSE loss only and MSE loss with physics-informed loss, respectively. The labeled data size in both cases is reduced to $1/32$ of the whole training data. Furthermore, Table 10 and Table 11 illustrate the results obtained when supervised models are trained solely with MSE loss, and MSE loss combined with physics-informed loss, respectively, using a labeled data size of $1/64$ of the whole training data. Note that training RCAN is not stable when using MSE plus physics-informed loss, so we only use MSE loss to train RCAN.

**Table 8:** Summary of the reconstruction performance, where supervised models are trained only with MSE loss. The size of labeled data is $1/32$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | RCAN | SRCNN | CFRF |
|---|---|---|---|---|---|---|---|
| MSE | 0.2286 | 0.1835 | 0.1017 | 0.0891 | 0.0864 | 0.0771 | **0.0751** |
| Equation Loss | 48.9912 | 1.4797 | 3.9548 | 2.6091 | 8.5349 | 0.9404 | **0.1984** |

**Table 9:** Summary of the reconstruction performance, where supervised models are trained with MSE loss and physics-informed loss. The size of labeled data is $1/32$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | SRCNN | CFRF |
|---|---|---|---|---|---|---|
| MSE | 0.2286 | 0.1835 | 0.1223 | 0.1367 | 0.0965 | **0.0964** |
| Equation Loss | 48.9912 | 1.4797 | 2.3207 | 1.6301 | 0.2500 | **0.1054** |

**Table 10:** Summary of the reconstruction performance, where supervised models are trained only with MSE loss. The size of labeled data is $1/64$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | RCAN | SRCNN | CFRF |
|---|---|---|---|---|---|---|---|
| MSE | 0.2286 | 0.1835 | 0.1078 | 0.0859 | 0.0892 | 0.0805 | **0.0784** |
| Equation Loss | 48.9912 | 1.4797 | 4.8715 | 2.6858 | 10.9180 | 0.9196 | **0.2115** |

**Table 11:** Summary of the reconstruction performance, where supervised models are trained with MSE loss and physics-informed loss. The size of labeled data is $1/64$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | SRCNN | CFRF |
|---|---|---|---|---|---|---|
| MSE | 0.2286 | 0.1835 | 0.1427 | 0.2106 | 0.0955 | **0.0954** |
| Equation Loss | 48.9912 | 1.4797 | 4.8581 | 3.3698 | 0.3316 | **0.1167** |

# G Speed Comparison

In this section, we compare simulation speed among our method, several baselines, and direct high-resolution simulation. Specifically, we simulate 250 trajectories of the 2D incompressible NS equation using a single NVIDIA A6000 GPU to compare the speed. The result is shown in Table 12. For HR simulation, we directly simulate 250 high-resolution trajectories. For mixed-resolution (MR) simulation + reconstruction, we follow the simulation protocol described in Section 5.1, where we simulate one high-resolution trajectory and the remaining trajectories are all low resolution. Then these low-resolution solutions are reconstructed to high resolution using our CFRF or baseline methods. From the result we can see that MR simulation + CFRF is $42\times$ faster than directly performing high-resolution simulation, improving the simulation time by more than 3.7 hours. While most remaining deep learning methods are faster than MR simulation + CFRF by several minutes due to the second unsupervised refinement stage employed by CFRF, we suggest this trade-off to be worthwhile for the improvement in physical consistency.

**Table 12:** Speed comparison of generating 250 HR trajectories among using direct HR simulation, our method, and other baselines. For deep learning methods, we perform mixed-resolution simulation first and then reconstruct HR solutions from LR solutions.

| Method | Total time |
|---|---|
| HR simulation | 3.8hrs |
| MR simulation + CFRF | 323s |
| MR simulation + SRCNN | 185s |
| MR simulation + RDN | 122s |
| MR simulation + FNO | 186s |
| MR simulation + RCAN | 413s |

# H More Results on Evaluating Visual Quality

To more comprehensively assess the visual quality of the reconstructed solutions, we adopt another two commonly used metrics in image super-resolution studies, namely the Peak Signal-to-Noise Ratio (PSNR) and Structured Similarity Indexing Measure (SSIM). These two metrics are better indicators of perceptual image quality and are more aligned with human visual perception. The results of these two metrics on three PDEs are shown in Tables 13 to 15. We can see that CFRF still outperforms baselines on these two metrics in most cases, except for the SSIM metric on the 2D diff-react equation, which demonstrates the effectiveness of our method. However, we must emphasize that unlike conventional image super resolution tasks, visual quality is not the single standard for evaluating the reconstruction of high-resolution numerical PDE solutions. Fluid fields are governed by physical laws described by a PDE, such as conservation of mass as given by the divergence-free constraint $\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0$ in the Navier-Stokes equations. Metrics such as MSE, PSNR, and SSIM fail to assess the degree to which the reconstructions adhere to the physical laws imposed by the governing PDE. Therefore, physical consistency metrics are crucial to a holistic evaluation of a PDE reconstruction method, for which we employ the equation loss. Our method can achieve up to 75% improvement in equation loss of reconstructed HR solutions, demonstrating that the reconstructions produced by CFRF not only have high visual quality, but also maintain physical consistency.

**Table 13:** Summary of the reconstruction performance for 2D incompressible NS equation, where supervised models are trained with MSE loss only. The size of labeled data is $1/16$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | RCAN | SRCNN | CFRF |
|---|---|---|---|---|---|---|---|
| PSNR | 27.6488 | 28.6030 | 31.3392 | 31.6674 | 31.8853 | 32.3865 | **32.5020** |
| SSIM | 0.5928 | 0.6519 | 0.7667 | 0.7695 | 0.7833 | 0.8031 | **0.8057** |

**Table 14:** Summary of the reconstruction performance for 2D shallow water equation on resolution on resolution 64 to 128, where supervised models are trained with MSE loss. The size of labeled data is $1/20$ of the whole training data. The result of RCAN is excluded as the training is not stable.

| Method | Nearest | Bicubic | FNO | RDN | SRCNN | CFRF |
|---|---|---|---|---|---|---|
| PSNR | 40.1929 | 42.4910 | 42.4872 | 40.0712 | 42.4872 | **42.9957** |
| SSIM | 0.9830 | 0.9896 | 0.9896 | 0.9847 | 0.9896 | **0.9900** |

**Table 15:** Summary of the reconstruction performance for 2D diffusion-reaction equation, where supervised models are trained with MSE loss. The size of labeled data is $1/20$ of the whole training data.

| Method | Nearest | Bicubic | FNO | RDN | RCAN | SRCNN | CFRF |
|---|---|---|---|---|---|---|---|
| PSNR | 29.4793 | 30.4170 | 31.8472 | 34.0815 | 33.7351 | 34.0898 | **34.0937** |
| SSIM | 0.4230 | 0.4633 | 0.5975 | **0.7933** | 0.7744 | 0.7862 | 0.7877 |

# I Reconstruction Visualization

We present visualizations of the reconstruction results obtained from our methods and compare them with those of baselines and ground truth. We need to emphasize that a good fluid flow reconstruction not only needs to achieve low MSE but also needs to be physically plausible. Although these predictions look similar (MSE improvement is marginal), it doesn't indicate how well the reconstructions satisfy the underlying equations. So we need to use both MSE and equation loss to evaluate the performance comprehensively. For example, on the 2D incompressible NS equation, our method outperforms baselines given the fact that our method achieves at most 75% improvement on physical consistency (equation loss) of reconstructions and marginally improves on MSE.

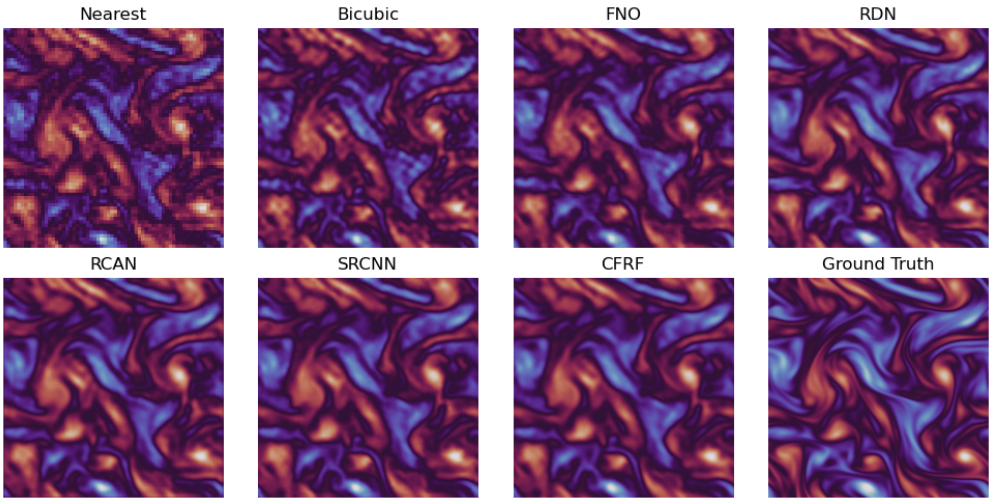


**Figure 4:** Visualization of reconstruction results for the 2D NS equations.

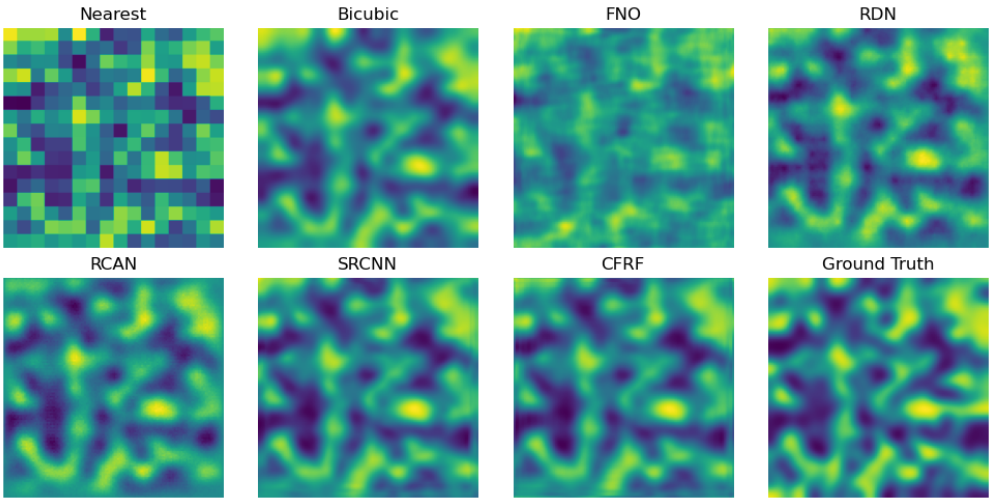


**Figure 5:** Visualization of reconstruction results for the 2D diffusion-reaction equations.

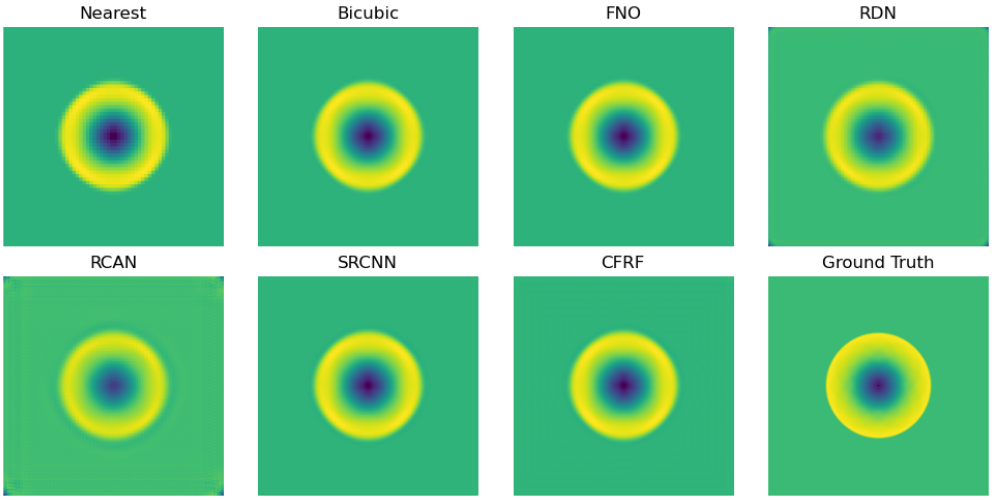


**Figure 6:** Visualization of reconstruction results for the 2D shallow water equations with resolution from 64 to 256.