Hierarchical Neural Operator Transformer with Learnable Frequency-aware Loss Prior for Arbitrary-scale Super-resolution

Xihaier Luo [1] **Xiaoning Qian** [1 2] **Byung-Jun Yoon** [1 2]

Abstract

In this work, we present an *arbitrary-scale* super-resolution (SR) method to enhance the resolution of scientific data, which often involves complex challenges such as continuity, multi-scale physics, and the intricacies of high-frequency signals. Grounded in operator learning, the proposed method is resolution-invariant. The core of our model is a hierarchical neural operator that leverages a Galerkin-type self-attention mechanism, enabling efficient learning of mappings between function spaces. Sinc filters are used to facilitate the information transfer across different levels in the hierarchy, thereby ensuring representation equivalence in the proposed neural operator. Additionally, we introduce a learnable prior structure that is derived from the spectral resizing of the input data. This loss prior is model-agnostic and is designed to dynamically adjust the weighting of pixel contributions, thereby balancing gradients effectively across the model. We conduct extensive experiments on diverse datasets from different domains and demonstrate consistent improvements compared to strong baselines, which consist of various state-of-the-art SR methods.

1. Introduction

Super-resolution (SR) plays a pivotal role in low-level vision tasks. The primary objective of SR is to transform blurred, fuzzy, and low-resolution images into clear, high-resolution images with enhanced visual perception. In recent years, deep learning has significantly advanced SR and has demonstrated promising performances in diverse domains beyond computer vision, including but not limited to medical imaging (Li et al., 2022), climate modeling (Reichstein et al., 2019), and remote sensing (Akiva et al., 2022). Nevertheless, existing deep learning-based SR methods often limit themselves to a *fixed-scale* (e.g., $\times 2$, $\times 3$, $\times 4$). The emergence of implicit neural representation (INR) in computer vision allows for continuous representation of complex 2D/3D objects and scenes (Xie et al., 2022). This development introduces opportunities for *arbitrary-scale* SR.

Challenges: Current *arbitrary-scale* SR methods, while capable of learning continuous representations from discretized data, face several challenges. 1) *Low-resolution Features*: The spatial resolution of extracted features is often inadequate for dense tasks such as image segmentation and regression. For example, using ResNet-50 on a 224×224 pixel input results in 7×7 deep features, showing a marked loss of resolution due to aggressive pooling (He et al., 2016). Even without resolution reduction, empirical evidence shows performance decline after flipping the feature map-a phenomenon termed flipping consistency decline (Song et al., 2023). 2) *Spectral Bias*: INRs are coordinate-based continuous functions usually parameterized by a multi-layer perceptron (MLP). The point-wise behavior of MLP in spatial dimensions poses challenges in learning high-frequency information, commonly known as spectral bias (Rahaman et al., 2019). This issue is particularly problematic when modeling scientific data, where the super-resolved predictions often appear over-diffused and fail to capture fine-scale details, such as small-scale vortices in turbulent flows (Fukami et al., 2019). 3) *Loss Function*: Most arbitrary-scale methods rely on per-pixel loss metrics (e.g., L1/L2 loss) (Liu et al., 2023). Training the model with a per-pixel L1 loss in a regression manner biases the reconstruction error towards an averaged output of all potential high-resolution images. Consequently, this often leads to blurry model predictions.

Solutions: For challenge 1), the key solution lies in upsampling deep features. Traditional spatial upsampling methods, such as transposed convolution, focus on local pixel attention and overlook global dependencies (Dumoulin & Visin, 2016). In contrast, Fourier domain upsampling supports global modeling. Therefore, we propose a hybrid approach

[1] Computational Science Intiative, Brookhaven National Laboratory, Upton, NY, USA [2] Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA. Correspondence to: Xihaier Luo <xluo@bnl.gov>, Byung-Jun Yoon <bjyoon@tamu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

that combines traditional convolution with spectral upsampling for enhanced performance. For challenge 2), we reformulate SR as operator learning (Kovachki et al., 2023). We replace the MLP-based inference network in an INR with a neural operator. This approach considers images as continuous functions rather than 2D pixel arrays, with each image instance representing a discretization of an underlying function. Here, we propose a hierarchical transformer as a neural operator for learning mappings between these function spaces. For challenge 3), we propose a spectral resizing-based loss prior. This prior is designed to adjust the contribution of each pixel to the overall loss within the image space. Such re-weighting redistributes gradients, thereby improving the SR model's ability to capture details across both high- and low-frequency regions.

The major contributions of our work include:

- We introduce a new hierarchical neural operator based on the transformer architecture with a Galerkin-type self-attention mechanism for *arbitrary-scale* super-resolution of scientific data.
- We devise a simple yet highly effective mechanism to construct a loss prior. It strategically adjusts pixel loss contributions during training and rebalances them in the next gradient updating step.
- We carry out extensive experiments to assess the effectiveness of our proposed method, showcasing its superiority compared to existing state-of-the-art (SOTA) SR approaches.

2. Background and Related Work

Super-resolution in many cases is a challenging ill-posed inverse problem. Its primary goal is to establish a mapping $\mathcal{F} : \mathbb{R}^{\mathbb{D}_a} \to \mathbb{R}^{\mathbb{D}_b}$ that transforms a given low-resolution (LR) input, denoted as $\boldsymbol{a}$, into a high-resolution (HR) output, denoted as $\boldsymbol{b}$, where $\mathbb{D}$ represents the discretization function and $\mathbb{D}(\boldsymbol{a})$ represents the discretized resolution of $\boldsymbol{a}$. The SR problem has traditionally been addressed by a spectrum of techniques, such as interpolation (Keys, 1981), neighbor embedding (Chang et al., 2004), sparse coding (Yang et al., 2010), and dictionary-based learning (Timofte et al., 2013). Overall, these approaches struggle to capture the intricate and nonlinear LR-to-HR transformation, often resulting in subpar super-resolved images, particularly in contexts involving high-wavenumbers (e.g. turbulence in Xie et al. (2018) and climate in Stengel et al. (2020)). On the other hand, the success of deep learning has brought about significant advancements in this field.

2.1. Deep Super-Resolution Models

Single-scale SR. Dong et al. (2014) were the first to introduce the CNN-based network to address SR challenges in natural images, achieving superior results compared to conventional methods. Following this, VDSR incorporated residual blocks (Kim et al., 2016), EDSR removed batch normalization layers, utilizing a residual scaling technique for training (Lim et al., 2017), RDN introduced the concept of dense feature fusion (Zhang et al., 2018), and SwinIR proposed bi-level feature extraction mechanism for local attention and cross-window interaction (Liang et al., 2021). These advancements collectively contributed to further enhancing the performance of super-resolution methods. Nonetheless, these approaches are confined to conducting upsampling with predefined factors, necessitating the training of separate models for each upsampling ratio, e.g. $\mathcal{F}_1$ for $\mathbb{D}(\boldsymbol{a}) \xrightarrow{s_1} \mathbb{D}(\boldsymbol{b})$ and $\mathcal{F}_2$ for $\mathbb{D}(\boldsymbol{a}) \xrightarrow{s_2} \mathbb{D}(\boldsymbol{b})$. This constraint restricts the practical applicability of these *single-scale* SR models (Liu et al., 2023).

Arbitrary-scale SR. Addressing this issue, a more recent and practical approach of *arbitrary-scale* super-resolution has emerged. MetaSR provides the first single-model solution for *arbitrary-scale* SR by predicting convolutional filter weights based on scale factors and coordinates, enabling adaptive filter weight prediction (Hu et al., 2019). In contrast to Meta-SR, LIIF employs an MLP as a local implicit function to predict RGB values based on queried HR image coordinates, extracted LR image features, and a cell size (Chen et al., 2021). LTE further introduced a local texture estimator that converts coordinates into Fourier domain data, enhancing the representational capacity of its local implicit function (Lee & Jin, 2022). More recently, LINF (Yao et al., 2023) and CiaoSR (Cao et al., 2023) have emerged to enhance LIIF's implicit neural representation. LINF focuses on learning local texture patch distributions, employing coordinate conditional normalizing flow for conditional signal generation. Alternatively, CiaoSR introduces an implicit attention network to determine ensemble weights for nearby local features. While effective, many *arbitrary-scale* SR methods heavily depend on continuous functions parameterized by MLPs. Unfortunately, MLPs are recognized for struggling with the learning of high-frequency functions, limiting their applicability in scientific domains characterized by complex, multiphysics-multiscale processes, such as weather data (Luo et al., 2023).

2.2. Neural Operator

Recently, neural operators have arisen as a promising approach for approximating functions (Kovachki et al., 2023). Unlike fully connected neural networks and convolutional neural networks, neural operators incorporate function space characteristics to guide network training, demonstrating superior performance in nonlinear fitting and maintaining invariance through discretization (Lu et al., 2021; Li et al., 2021; Hao et al., 2023; Tran et al., 2023). SRNO exemplifies the application of neural operators in super-resolution tasks,

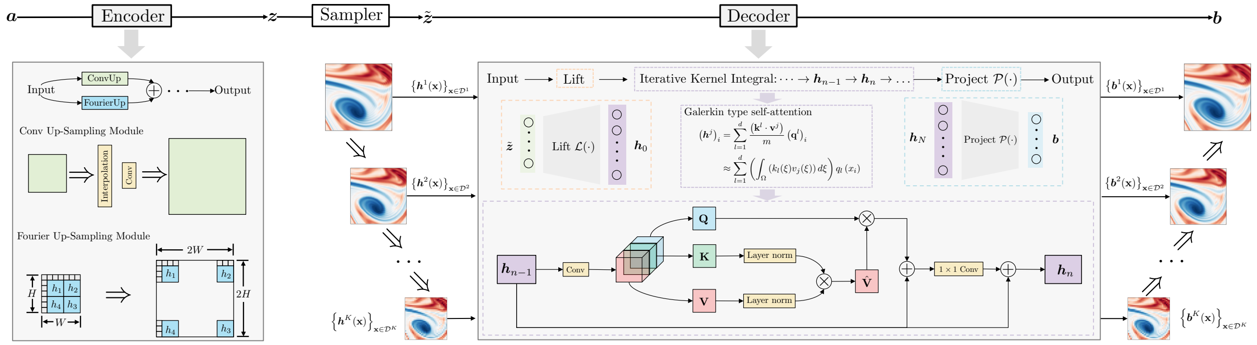
a
Encoder
z
Sampler
$\tilde{z}$
Decoder
b
Input
ConvUp
FourierUp
Output
Conv Up-Sampling Module
Interpolation
Conv
Fourier Up-Sampling Module
$2W$
H
h_1
h_2
h_4
h_3
W
$2H$
$\{h^1(\mathbf{x})\}_{\mathbf{x}\in\mathcal{D}^1}$
$\{h^2(\mathbf{x})\}_{\mathbf{x}\in\mathcal{D}^2}$
$\{h^K(\mathbf{x})\}_{\mathbf{x}\in\mathcal{D}^K}$
Input
Lift
Iterative Kernel Integral: $\cdots \to h_{n-1} \to h_n \to \ldots$
Project $\mathcal{P}(\cdot)$
Output
$\tilde{z}$
Lift $\mathcal{L}(\cdot)$
h_0
Galerkin type self-attention
$(h^j)_i = \sum_{l=1}^{d} \frac{(\mathbf{k}^l \cdot \mathbf{v}^j)}{m} (\mathbf{q}^l)_i$
$\approx \sum_{l=1}^{d} \left(\int_\Omega (k_l(\xi) v_j(\xi))\, d\xi \right) q_l(x_i)$
h_N
Project $\mathcal{P}(\cdot)$
b
h_{n-1}
Conv
Q
K
V
Layer norm
Layer norm
$\hat{\mathbf{V}}$
1×1 Conv
h_n
$\{b^1(\mathbf{x})\}_{\mathbf{x}\in\mathcal{D}^1}$
$\{b^2(\mathbf{x})\}_{\mathbf{x}\in\mathcal{D}^2}$
$\{b^K(\mathbf{x})\}_{\mathbf{x}\in\mathcal{D}^K}$

Figure 1. Overview of the **Hi**erarchical **N**eural **O**perator **T**ransform**E**r (HiNOTE). HiNOTE features a structured architecture comprising three key components: Firstly, an encoder designed for upsampling deep features; secondly, a sampler tasked with rendering a specific set of features; and thirdly, a decoder capable of making inferences at various arbitrary points within the domain.

employing a linear attention operator for Petrov-Galerkin projection (Wei & Zhang, 2023). DFNO is another instance employing a Fourier neural operator for downscaling climate variables (Yang et al., 2023). This model undergoes training with data featuring a modest upsampling factor and subsequently demonstrates the ability to perform zero-shot downscaling to arbitrary, unseen high resolutions.

3. Method

To achieve *arbitrary-scale* SR, we introduce **Hi**erarchical **N**eural **O**perator **T**ransform**E**r (HiNOTE), featuring a hybrid upsampling module and a frequency-aware loss prior (See Fig. 1). These designs enable accurate learning of the continuous representation of the underlying data.

3.1. Problem Statement

Problem Reformulation. In the context of single-scale SR, different methods aim to establish a mapping $f : \mathbb{R}^{d_a} \to \mathbb{R}^{d_b}$ from low-resolution input $\boldsymbol{a} \in \mathbb{R}^{d_a}$ to high-resolution output $\boldsymbol{b} \in \mathbb{R}^{d_b}$ with $d_a < d_b$. This formulation imposes a constraint fixing the input/output resolution at d_a/d_b. To overcome this limitation, we propose to learn a mapping from an infinite-dimensional function space to another infinite-dimensional function space $\mathcal{G}^\dagger : \mathcal{A} \to \mathcal{B}$. Here, $\mathcal{A} = \mathcal{A}(D; \mathbb{R}^{d_a})$ and $\mathcal{B} = \mathcal{B}(D; \mathbb{R}^{d_b})$ are separable Banach spaces of functions taking values in $\mathbb{R}^{d_a}$ and $\mathbb{R}^{d_b}$ respectively (Kovachki et al., 2023). This approach models observed digital data (dynamic fields, images of 2D pixel arrays, or 1D acoustic signals) by continuous function representations, treating corresponding data as observed measurements in discretized domains of the underlying functions. It allows for obtaining arbitrarily low-resolution inputs and high-resolution outputs by evaluating $\boldsymbol{a}$ and $\boldsymbol{b}$ at numerous points within D. For instance, $\boldsymbol{a}(\mathbb{D}) = \{a(x_1), ..., a(x_n)\}$, where $\mathbb{D} = \{x_1, \ldots, x_n\} \subset D$ is a n-point discretization of the domain D. For simplicity in the following discussion, $\boldsymbol{a}$ and $\boldsymbol{b}$ will be used unless stated otherwise.

Optimization Goal. This new formulation based on continuous function representations enables arbitrary-scale SR: Given observations $\{\boldsymbol{a}^{(j)}, \boldsymbol{b}^{(j)}\}_{j=1}^{N}$, where $\boldsymbol{a}^{(j)}$ is an i.i.d. low-resolution sample and $\boldsymbol{b}^{(j)} = \mathcal{G}^\dagger(\boldsymbol{a}^{(j)})$ represents the high-resolution counterpart, we seek to build a parametric map to approximate $\mathcal{G}^\dagger$:

$$\mathcal{G}_\theta : \mathcal{A} \times \Theta \to \mathcal{B}, \quad \theta \in \Theta \tag{1}$$

for some finite-dimensional parameter space Θ by choosing $\theta^\dagger \in \Theta$ so that $\mathcal{G}_{\theta^\dagger} \approx \mathcal{G}^\dagger$.

3.2. Overview of Model Architecture

Following the defined problem, to achieve the goal of learning the mapping from the function space $\mathcal{A}$ to $\mathcal{B}$, we design $\mathcal{G}_\theta$ as a composition of a hybrid upsampling-based encoder, E_φ, and more importantly, a new hierarchical neural operator transformer-based decoder, D_ϕ, stacked before and after a parameter-free sampler $\mathcal{S}$ (Fig. 1):

$$\mathcal{G}_\theta := D_\phi \circ \mathcal{S} \circ E_\varphi \tag{2}$$

1. The encoder $E_\varphi : \mathcal{A}(D; \mathbb{R}^{d_a}) \to \mathcal{Z}(D; \mathbb{R}^{d_z})$ is a functional operator parameterized by φ. It maps the input data $\boldsymbol{a}^{(j)}$ to a condensed representation, i.e., a feature vector $\boldsymbol{z}^{(j)} = E(\boldsymbol{a}^{(j)}; \varphi)$. We incorporate a hybrid upsampling module into model E_φ, designed to effectively capture both spatial and spectral information as well as local and global details.

2. The sampler $\mathcal{S}$ is parameter-free and acts as an upsampling module $\mathcal{S} : \mathcal{Z}(D; \mathbb{R}^{d_z}) \to \mathcal{Z}(D; \mathbb{R}^{d_b})$. It aligns the encoder output size d_z with any arbitrarily chosen high-resolution discretization d_b.

3. The decoder $D_\phi : \mathcal{Z}(D; \mathbb{R}^{d_b}) \to \mathcal{B}(D; \mathbb{R}^{d_b})$ is a functional operator parameterized by ϕ. It enables the generation of super-resolved outputs at any specified resolution during the inference $\boldsymbol{b}^{(j)} = D(\boldsymbol{z}^{(j)}; \phi)$.

3.3. Encoder

The encoder is engineered to extract a series of deep features from the input data. However, these deep features often lack the spatial resolution needed for tasks like segmentation and depth prediction. This spatial insufficiency is amplified in the SR setting, where every pixel from low-resolution input is important. Common spatial up-sampling methods in convolutional neural networks rely on local pixel attention, limiting global dependency exploration. In contrast, the Fourier domain aligns with global modeling principles, as per the spectral convolution theorem (Brigham, 1988). To address this, we introduce a hybrid upsampling module (See the encoder details in Fig. 1). It merges conventional convolution-based upsampling with a recently developed Fourier upsampling module (Yu et al., 2022). Our proposed hybrid upsampling module effectively captures global features and preserves overall structural integrity while also benefiting from the local context awareness inherent in convolutional operations. An additional enhancement we introduced pertains to the placement of the upsampling module. While conventional SR models typically position the upsampling layer towards the end of the network. We propose that refining feature maps at the network's onset is more advantageous (See Fig. 8).

3.4. Sampler

The sampler operator utilizes a patch-based interpolation scheme to transform a discrete feature vector from the existing resolution d_z to any target resolution d_b. Specifically, this involves two steps: *feature map rendering* and *patch ensemble*, as shown in Fig. 2.

Feature Map Rendering. For rendering a new feature map from a discrete feature map with size $h \times w$, we assume feature vectors are evenly distributed in a 2D domain $[-1, 1] \times [-1, 1]$. Dividing this into $h \times w$ regions, each cell $(\Delta x \times \Delta y)$ in the feature map $\boldsymbol{z}^{(j)}$ is associated with absolute central coordinates (x, y) in the corresponding region. Given an arbitrary point at $(x^\star, y^\star)$, we first calculate the distances to the nearest four neighboring central coordinates (x_i, y_i), where $i \in \{00, 01, 10, 11\}$ represents the nearest feature vectors in the top-left, top-right, bottom-left, and bottom-right sub-spaces:

$$x'_i = \frac{(x^\star - x_i) \cdot \Delta x}{2}, \quad y'_i = \frac{(y^\star - y_i) \cdot \Delta y}{2}. \tag{3}$$

We then normalize the feature vectors $\boldsymbol{z}^{(j)}(x_i, y_i)$ based on the area of the rectangle a_i between the query point and its nearest feature vector's diagonal counterpart:

$$\boldsymbol{z}^{(j)}(x^\star, y^\star) = \boldsymbol{z}^{(j)}(x_i, y_i) \cdot a_i(x'_i, y'_i) \, / \, (\Delta x \cdot \Delta y). \tag{4}$$

Patch Ensemble. Instead of merging normalized feature vectors, we propose feature ensembling. These normalized features, along with positional information, are fused in the decoder $D_\phi(\boldsymbol{z}^{(j)})$ to enhance local feature ensembling:

$$\tilde{\boldsymbol{z}}^{(j)} = \{\boldsymbol{z}^{(j)}(x^\star, y^\star), x_i, y_i, \Delta x, \Delta y\}_i. \tag{5}$$

Compared to direct weighted summation $\sum_i a_i \boldsymbol{z}_i^{(j)}$, which relies only on local feature coordinates (Chen et al., 2021), fusing continuous feature maps $\tilde{\boldsymbol{z}}^{(j)}$ in the decoder allows the model to leverage more information from local features.

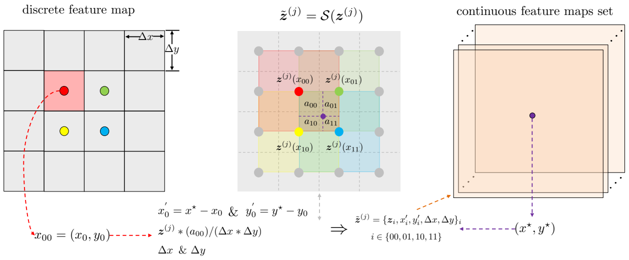
discrete feature map
Δx
Δy
$\tilde{\boldsymbol{z}}^{(j)} = \mathcal{S}(\boldsymbol{z}^{(j)})$
$\boldsymbol{z}^{(j)}(x_{00})$
$\boldsymbol{z}^{(j)}(x_{01})$
a_{00}
a_{01}
a_{10}
a_{11}
$\boldsymbol{z}^{(j)}(x_{10})$
$\boldsymbol{z}^{(j)}(x_{11})$
continuous feature maps set
$x_{00} = (x_0, y_0)$
$x'_0 = x^\star - x_0$ & $y'_0 = y^\star - y_0$
$\boldsymbol{z}^{(j)} * (a_{00})/(\Delta x * \Delta y)$
Δx & Δy
$\Rightarrow$
$\tilde{\boldsymbol{z}}^{(j)} = \{\boldsymbol{z}_i, x'_i, y'_i, \Delta x, \Delta y\}_i$
$i \in \{00, 01, 10, 11\}$
$(x^\star, y^\star)$

Figure 2. Illustration of the parameter-free sampler. It samples arbitrary resolutions from feature maps extracted by the encoder and combines the positional information of the grid points.

3.5. Decoder

The decoder D_ϕ is a functional operator parameterized by a neural operator. It typically comprises three components:

- Lifting: The input $\tilde{\boldsymbol{z}} \in \tilde{\mathcal{Z}}$ undergoes lifting to its first hidden representation $\boldsymbol{h}_0 = \mathcal{L}(\tilde{\boldsymbol{z}})$ using a point-wise function $\mathbb{R}^{d_{\tilde{z}}} \rightarrow \mathbb{R}^{d_{h_0}}$. This lifting operation is achieved by a fully connected neural network with dimensions $d_{\tilde{z}} < d_{h_0}$.
- Iterative Kernel Integral: For $n = 0, \ldots, N-1$, each hidden representation evolves through an iterative kernel integral approximation $(\mathcal{K}_n(\boldsymbol{h}_n))(x) = \int_D \kappa^{(n)}(x, y)\boldsymbol{h}_n(y)\mathrm{d}y, \forall x \in D$, where the kernel matrix $\kappa^{(n)} : \mathbb{R}^{d+d} \rightarrow \mathbb{R}^{d_h \times d_h}$ is a neural network.
- Projection: The output $\boldsymbol{u}$ is the projection of the last hidden representation's output, $\boldsymbol{b} = \mathcal{P}(\boldsymbol{h}_N)$, using a local transformation $\mathbb{R}^{d_{h_N}} \rightarrow \mathbb{R}^{d_b}$. Similar to the lifting step, this is performed by a fully connected neural network, typically with $d_{h_N} > d_b$.

The core of the decoder D_ϕ is the kernel integral, enabled by a transformer-based neural operator. This choice is driven by two primary factors:

1. *Flexibility in Input and Output Sizes*: Transformers are adept at handling variable input and output data sizes (Vaswani et al., 2017). This adaptability is essential for our *arbitrary-scale* SR tasks, where the domain size or discretization level may vary.

2. *Ability to Capture Long-Range Dependencies*: Transformers excel at capturing long-range dependencies in data, a critical feature for scientific applications like global weather pattern prediction, where interactions between distant variables are essential for accurate predictions (Gao et al., 2022).

We complete the design of HiNote, our proposed framework. HiNOTE adopts a U-Net architecture with bandlimited functions as both inputs and outputs. This design renders HiNOTE a representationally equivalent neural operator in terms of aliasing errors (Karras et al., 2021; Raonic et al., 2023). For the kernel integral, HiNOTE incorporates a Galerkin-type self-attention mechanism, which effectively reduces computational complexity from quadratic to linear (Cao, 2021).

3.5.1. Hierarchical Architecture

Depicted in Fig. 2, we use a parameterized downsample layer to derive deep representations $\{\{\boldsymbol{h}^k(\mathbf{x})\}_{\mathbf{x}\in\mathcal{D}^k}\}_{k=1}^{K}$ across K scales. These representations aggregate local observations with learnable parameters, with $\{\boldsymbol{h}^1(\mathbf{x})\}_{\mathbf{x}\in\mathcal{D}^1}$ representing the finest resolution. Traditional downsampling methods, such as affine linear transformations combined with a nonlinear activation, often lead to aliasing errors by not adhering to the band-limits of the underlying function space (Raonic et al., 2023). To tackle this issue, we first upsample the input function beyond its frequency bandwidth; after the activation function, the signal is downsampled.

Upsampling The process involves initially augmenting the number of samples in the signal. For instance, consider upsampling a single channel signal $\boldsymbol{h}$ in $\mathbb{R}^{n_x \times n_y}$ to $\boldsymbol{h}'$ in $\mathbb{R}^{n_x N \times n_y N}$. This is achieved by interspersing each pair of signal samples with $N-1$ zero-valued samples:

$$\boldsymbol{h}^{\uparrow}[i,j] = \mathbb{I}_{n_x}(i)\cdot\mathbb{I}_{n_y}(j)\cdot\boldsymbol{h}[i \bmod n_x, j \bmod n_y], \quad (6)$$

where $i = 1 \ldots N \cdot n_x$ and $j = 1 \ldots N \cdot n_y$. Subsequently, the upsampled signal is convolved with an interpolation filter, which serves to remove high-frequency components.

Downsampling We utilize a sinc-based low-pass filter and execute downsampling post-nonlinear activation:

$$\boldsymbol{h}^{\downarrow} = \left(\frac{w_{out}}{w_{in}}\right)^2 \left(f_{w_{out}} \star \boldsymbol{h}\right)(x), \quad \forall x \in D \quad (7)$$

where $\star$ is the convolution operation and $f_{w_{out}}(x_0, x_1) = \operatorname{sinc}(2wx_0)\cdot\operatorname{sinc}(2wx_1)$ is a sinc-based low-pass filter.

3.5.2. Galerkin-type Self-attention

Building upon the proposed hierarchical network structure, the next step is to conduct iterative kernel integral at each hierarchical level. Based on Nyström approximation theory, there is a similarity between the attention matrix in transformers and an integral kernel (Kovachki et al., 2023). More precisely, dot-product attention can be interpreted as an approximation of an integral transform using a non-symmetric, trainable kernel function (Cao, 2021; Li et al., 2023). In this study, we adopt the perspective of learnable kernel integrals for attention, treating each channel in the hidden feature map as a sample from a distinct function on the discretization grid. Omitting the layer index, let $\boldsymbol{h} = (h(x_1), \ldots, h(x_m))^T \in \mathbb{R}^{m\times d_h}$ denote evaluations in the iterative kernel integral. Consider matrices $\{\mathbf{Q}, \mathbf{K}, \mathbf{V}\} \in \mathbb{R}^{m\times d_h}$ as query/key/value matrices. The columns of $\mathbf{Q}/\mathbf{K}/\mathbf{V}$ contain vector representations of learned basis functions, spanning subspaces in latent representation Hilbert spaces $\mathbf{Z} = \frac{1}{m}\mathbf{Q}\left(\widehat{\mathbf{K}}^T\widehat{\mathbf{V}}\right)$, where $\widehat{\cdot}$ denotes a column-wise normalized matrix. For instance, $\widehat{\mathbf{V}}_{ij}$ (also the i-th element of the j-th column vector: $(\mathbf{v}^j)_i$) is the evaluation of the j-th basis function on the i-th grid point x_i, i.e., $\widehat{\mathbf{V}}_{ij} = v_j(x_i)$. Similarly, for matrices $\mathbf{Q}, \mathbf{K}$, each column represents the sampling of basis functions $q_j(\cdot)$ and $k_j(\cdot)$, respectively. Leveraging this interpretation of learnable bases, we can employ the Monte-Carlo method,

$$\left(\boldsymbol{h}^j\right)_i = \sum_{l=1}^{d} \frac{\left(\mathbf{k}^l \cdot \mathbf{v}^j\right)}{m} \left(\mathbf{q}^l\right)_i, \quad (8)$$

to approximate the kernel integral. Hence, the kernel integral is iteratively executed through Galerkin-type self-attention (Cao, 2021). In contrast to standard attention, Galerkin-type self-attention reduces the quadratic complexity from $\mathcal{O}(m^2 d_h)$ to a linear complexity of $\mathcal{O}(m d_h^2)$.

3.6. Training

Though our model $\mathcal{G}_\theta$ integrates three components E_φ, $\mathcal{S}$ and D_ϕ, it is trained jointly in an end-to-end fashion. While DNN-based model excels in capturing complex signals, accurately approximating high-frequency details remains challenging (Sitzmann et al., 2020).

To address this, we introduce a frequency-aware loss prior in the corresponding discretized domain:

$$\boldsymbol{p} = |\mathcal{G}(\boldsymbol{a}) - \mathcal{R}(\boldsymbol{a})|, \quad \boldsymbol{p} \in \mathbb{R}^{d_b}, \quad (9)$$

where $\mathcal{R} : \mathbb{R}^{d_a} \to \mathbb{R}^{d_b}$ is the spectral resizing function. Next, we rescale $\boldsymbol{p}$ using min-max normalization $n(\boldsymbol{p}) = (\boldsymbol{p} - \min(\boldsymbol{p}))/(\max(\boldsymbol{p}) - \min(\boldsymbol{p}))$. Applying the exponential function to $n(\boldsymbol{p})$ yields non-zero weights $\mathbf{W} = \exp(n(\boldsymbol{v}))$. Similar to Jiang et al. (2021) and Gou et al. (2023), we introduce hyperparameters α and β to enhance the expressibility and controllability of the weights

$$\mathbf{W}(\boldsymbol{p}; \alpha, \beta) = \alpha \cdot \exp(\beta \cdot n(\boldsymbol{p})). \quad (10)$$

During training, the weighting matrix $\mathbf{W}(\boldsymbol{p}; \alpha, \beta)$ is utilized to adjust the weights. This adjustment can be implemented either as a penalty term in the loss function or through a two-step training approach. Due to limited space, detailed implementations with this learnable frequency-aware loss prior are provided in Appendix B.2.2.

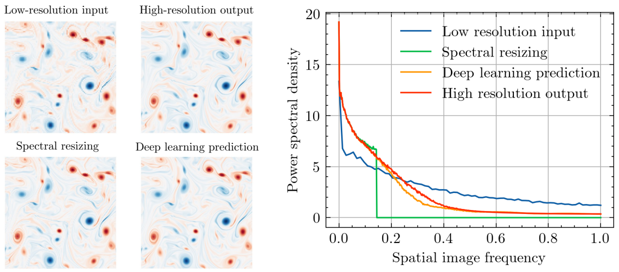

Figure 3. Distinguishing between pixels of different frequency regions in image space poses a challenge. Deep learning predictions often show high visual perception metrics when compared with target HR images (See the left representations). We analyze the images in the frequency domain and observe that the power spectra of HR images and those produced by deep learning models begin to diverge at a certain frequency (e.g., 0.2 in this example). To identify this frequency, spectral resizing is applied to LR inputs, revealing a clear demarcation in frequency regions. This demarcation aligns with the frequency divergence. Building on this, we introduce a static structure prior, created by subtracting low-frequency signals (obtained via spectral resizing) from deep learning predictions in the image space. This prior is then utilized to refine and enhance the network training process.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our proposed method against baseline methods using four diverse datasets. Detailed information about the generation and preprocessing of these four datasets can be found in Appendix A.

- **Turbulence Flow**: We explore two-dimensional Kraichnan turbulence in a doubly periodic square domain spanning $[0, 2\pi]^2$. The Navier-Stokes equation is solved through direct numerical simulation to generate the required data (Pawar et al., 2023).
- **Global Weather Pattern**: We utilize weather data encompassing ERA5 reanalysis data, high-resolution simulated surface temperature at 2 meters, kinetic energy at 10 meters above the surface, and total column water vapor (Hersbach et al., 2020).
- **SEVIR**: We utilize the Storm EVent ImagRy (SEVIR) dataset, which comprises a large, curated collection of labeled examples. This dataset encompasses various weather phenomena, including thunderstorms, convective systems, and related events (Veillette et al., 2020).
- **MRI**: The magnetic resonance imaging (MRI) dataset includes a variety of snapshots from multiple sources. Our primary focus is on brain scans that utilize a horizontal sampling mask (Jalal et al., 2021).

Baselines. We benchmark HiNOTE against several well-acknowledged and advanced models, encompassing five *single-scale* SR baselines: SRCNN (Dong et al., 2015), ESPCN (Shi et al., 2016), EDSR (Lim et al., 2017), WDSR (Yu et al., 2018), SwinIR (Liang et al., 2021), and five *arbitrary-scale* SR baselines: MetaSR (Hu et al., 2019), LIIF (Chen et al., 2021), LTE (Lee & Jin, 2022), DFNO (Yang et al., 2023), and SRNO (Wei & Zhang, 2023). Due to space constraints, the configuration details for each baseline model are provided in Appendix B.

Evaluation Protocol. We utilize an A100 GPU with 48GB capacity. Training models directly on HR data poses significant challenges due to memory constraints. Consequently, we employ a strategy of randomly cropping each high-resolution snapshot into smaller segments for training purposes. To ensure fairness in comparison, all methods are trained using the L1 loss function for 300 epochs. This training employs the AdamW optimizer (Loshchilov & Hutter, 2019), with the initial learning rate determined through hyperparameter tuning. Due to the varying sizes of each model and the limitations imposed by GPU memory, batch sizes may differ (Details are available in Appendix B.1). For the evaluation of model performance, we utilize mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) as our primary metrics.

4.2. Main Results

Single-scale SR Performance. In a single-scale SR task (32×32 to 128×128), various models were trained and their performance compared, as shown in Table 1. Key observations include: (1) HiNOTE consistently outperforms other models in all SR tasks. It shows notable improvements in weather data, e.g., a 13.29% enhancement on Temperature, a 34.44% improvement on Kinetic Energy and a 3.09% improvement on Water Vapor. 'Improvement' here refers to the relative error reduction compared to the second-best model. (2) Other baselines also show reasonable performance. SwinIR, using a hierarchical transformer architecture and shifted window strategy, is the most competitive across datasets but has a higher computational demand, with 2.1 million parameters compared to HiNOTE's 1.5 million. HiNOTE, with its Galerkin-type self-attention, offers reduced computational needs. (3) Convolutional neural network (CNN)-based models like EDSR and WDSR outperform transformer-based models like SwinIR in turbulence data. On the other hand, HiNOTE leverages a UNet-based hierarchical model architecture, enabling it to effectively learn datasets characterized by patterns spanning various

Table 1. Quantitative comparison results for the ×4 super-resolution task. Superior performance is denoted by a smaller MSE and higher PSNR/SSIM values. Light red indicates the top performer, while light green represents the second-best performer among all baselines and our method for each metric.

	Turbulence			Temperature			Kinetic Energy			Water Vapor		
Model	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
							single-scale baselines					
SRCNN	1.196e-4	39.138	0.972	5.206e-5	42.690	0.980	3.595e-4	33.997	0.924	9.992e-5	39.912	0.975
ESPCN	1.587e-5	47.979	0.993	2.732e-5	45.472	0.985	1.896e-4	36.172	0.956	4.257e-5	43.417	0.985
EDSR	7.303e-6	51.305	0.997	2.831e-5	45.317	0.985	1.963e-4	36.465	0.957	4.890e-5	42.815	0.983
WDSR	6.270e-6	51.898	0.997	2.441e-5	45.984	0.986	1.769e-4	36.792	0.955	4.837e-5	42.662	0.983
SwinIR	2.789e-5	45.531	0.997	2.633e-5	45.722	0.986	1.736e-4	36.874	0.957	3.209e-5	44.449	0.986
							arbitrary-scale baselines					
MetaSR	7.988e-5	40.879	0.976	5.270e-5	42.639	0.978	2.199e-4	35.751	0.946	8.489e-5	40.564	0.977
LIIF	2.818e-5	45.482	0.988	2.969e-5	45.273	0.984	1.888e-4	35.944	0.943	5.460e-5	42.336	0.980
LTE	1.032e-5	48.792	0.993	2.796e-5	45.486	0.985	1.663e-4	36.313	0.951	3.266e-5	44.052	0.986
DFNO	2.147e-4	36.543	0.985	1.905e-4	37.048	0.980	2.914e-4	34.397	0.936	3.967e-4	33.867	0.944
SRNO	9.083e-6	50.403	0.996	2.287e-5	46.355	0.987	2.083e-4	36.635	0.949	3.446e-5	44.299	0.986
HiNOTE	6.112e-6	51.903	0.997	1.983e-5	46.424	0.987	1.138e-4	36.995	0.958	3.112e-5	44.501	0.987

Table 2. Quantitative comparison results for the *arbitrary-scale* SR tasks. For each metric, light red indicates the top performer and light green represents the second-best performer.

	×4.6			×8.2			×15.7			×32		
Model	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
							Interpolation Methods					
Bilinear	8.625e-5	39.177	0.973	3.141e-4	33.5634	0.936	8.053e-4	29.475	0.913	1.703e-3	26.223	0.903
Bicubic	9.323e-5	38.839	0.971	3.518e-4	33.0714	0.931	9.126e-4	28.932	0.909	1.960e-3	25.612	0.902
Nearest	1.341e-4	37.258	0.934	4.716e-4	31.7989	0.859	1.179e-3	27.817	0.831	2.454e-3	24.635	0.841
							Deep Learning Methods					
MetaSR	9.274e-5	40.295	0.974	4.043e-4	33.900	0.904	1.227e-3	29.077	0.842	2.716e-3	25.628	0.806
LIIF	2.818e-5	45.482	0.988	2.671e-4	35.715	0.940	9.731e-4	30.101	0.886	2.283e-3	26.397	0.874
LTE	7.860e-6	50.985	0.996	2.080e-4	36.759	0.951	1.033e-3	29.796	0.875	2.466e-3	26.018	0.844
DFNO	2.364e-4	36.102	0.984	4.339e-4	33.489	0.934	1.188e-3	29.112	0.878	2.412e-3	26.038	0.854
SRNO	9.272e-6	50.081	0.995	1.634e-4	37.852	0.959	8.008e-4	30.759	0.912	2.127e-3	26.708	0.879
HiNOTE	7.121e-6	51.225	0.996	8.906e-5	40.002	0.976	7.361e-4	32.164	0.920	2.041e-3	26.772	0.891

scales. These finding aligns with the nature of turbulence, a small-scale phenomenon, and the strengths of CNNs in capturing fine details, contrasting with attention blocks' effectiveness in larger-scale weather patterns (Gao et al., 2022). Complete results for the additional two datasets can be found in the Appendix B.3 and B.4.

Arbitrary-scale SR Performance. Table.2 presents a quantitative comparison of HiNOTE with SOTA arbitrary-scale SR models and traditional interpolation methods. For training, all deep learning models employ an upsampling ratio s_i randomly drawn from a uniform distribution $\mathcal{U}[1, 4]$. We tested s_i ranging from a moderate extrapolation of 4.6 to an extreme ratio of 32, noting that these ratios were not included in training. The results reveal two key findings: (1) HiNOTE surpasses current SOTA methods, achieving an average improvement of 20.11% over the second-best model, SRNO, across various upsampling ratios; (2) As s_i increases, the relative performance of deep learning models compared to classical interpolation methods diminishes. Notably, at $s_i = 32$, bilinear interpolation often outperforms deep learning in many metrics. This suggests that the reliable extrapolation range for deep learning models is approximately 16, considering their training on a maximum upsampling ratio of 4. Including larger upsampling ratios in training could potentially extend this range.

Showcases. Fig.4 presents qualitative comparisons between the HiNOTE and other leading arbitrary-scale SR methods, specifically LIIF and SRNO. An expanded qualitative comparison is detailed in the Appendix B.3. The results illustrate HiNOTE's ability to generate super-resolved images with notably sharper textures compared to these methods. For example, in the second row, which displays the absolute error between the target and the predictions, HiNOTE demonstrates superior performance to LIIF, as evidenced by the significantly lower error margins. Against SRNO, the previously best-performing arbitrary-scale SR method, HiNOTE shows marked improvements, especially noticeable in the errors pertaining to ocean areas in the images. This enhanced performance further substantiates the effectiveness of HiNOTE's hierarchical structure in accurately

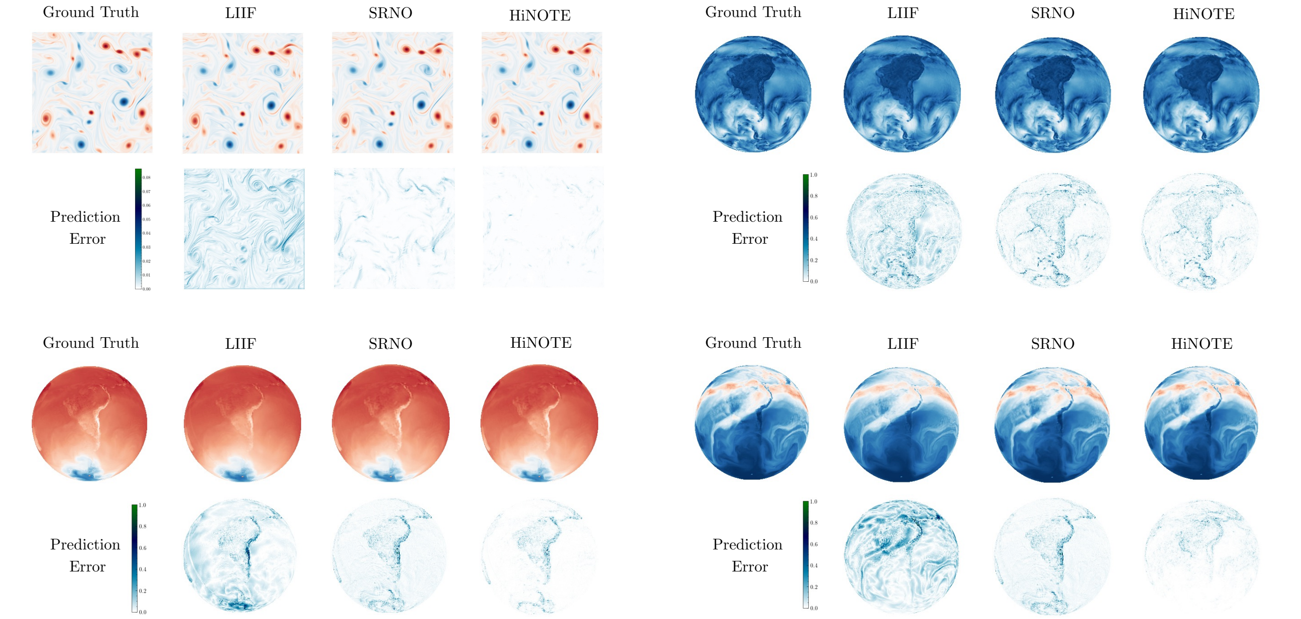

Figure 4. Qualitative comparison with state-of-the-art (SOTA) methods for arbitrary-scale SR. Top-left: turbulence flow; Top-right: kinetic energy; Bottom-left: temperature; and Bottom-right: water vapor.

modeling multi-scale data, such as climate-related imagery.

Continuous Representation. Fig. 5 presents a comparative analysis of arbitrary SR methods: the bilinear interpolation-based approach versus the deep learning-based HiNOTE. Each subplot displays the predicted high-resolution output at different upsampling ratios using turbulence flow data, with specific focus on regions abundant in high-frequency structures (highlighted by rectangular boxes). The comparison reveals the limitations of bilinear interpolation, particularly its restricted context awareness and reliance on neighboring pixels, resulting in imprecise high-frequency detail representation. This deficiency is more pronounced at higher upsampling ratios (e.g., ×32), producing images that are blurry and lack detail. Conversely, HiNOTE effectively reconstructs fine structures, underscoring the significance of global integral kernels in capturing overall structures.

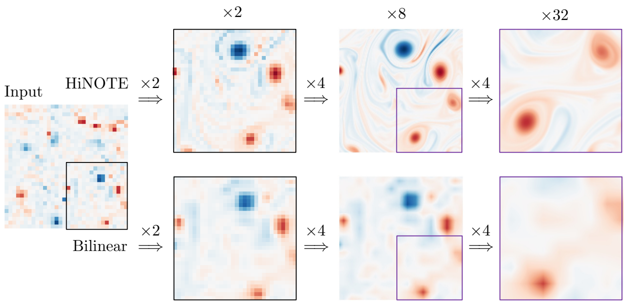

Figure 5. Qualitative demonstration of continuous representation learning: model performance evaluated on two instances randomly chosen from the test dataset, across various upsampling ratios.

4.3. Ablation Study

Necessity of Refining Features. To investigate the importance of refining deep features, we trained HiNOTE across various upsampling ratios. Next, performance was evaluated using a test dataset with an upsampling ratio fixed at 8. The results reveal that enhancing the spatial resolution of feature maps improves performance in tasks such as SR. Notably, omitting up-sampling resulted in the lowest performance (×1). The optimal up-sampling ratio varies with the problem. In this case, a model trained with a fourfold increase outperforms a twofold increase by 4.16%.

Table 3. Experimental results for different up-sampling ratios.

Up-sampling ratio	×1	×2	×4
MSE	9.869e-5	9.307e-5	8.919e-5

Efficiency of the Upsampling Module. It is critical to note that adding an upsampling layer at the end of the network increases the computational cost. Traditionally, a SR model requires a deep encoder to extract complex, abstract features through upsampling at the end of the network. In contrast, HiNOTE incorporates the iterative kernel integral as a central feature, allowing for fewer channels thanks to our proposed patch ensemble approach. Consequently, our encoder requires fewer layers, significantly reducing the overall computational demands. We have performed a comparative analysis to assess the computational effects of including versus excluding the upsampling module. The results, detailed below, demonstrate a minimal increase in computational cost due to the efficient design of our model.

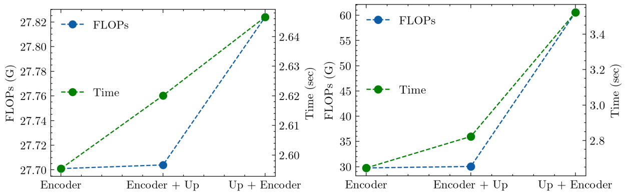

Figure 6. Efficiency comparison of three models: (1) without upsampling, (2) with upsampling after encoder, and (3) with upsampling before encoder.

Ablations on Self-attention Mechanisms. The primary motivation of employing Galerkin-type self-attention is to decrease computational complexity. Among various self-attention mechanisms that also reduce complexity, Galerkin-type self-attention is preferred due to its origin in operator-based learning problems, aligning closely with our model design. Here, we have conducted an ablation study of different self-attention mechanisms: Vanilla (Vaswani et al., 2017), FAVOR (Choromanski et al., 2020), and ProbSparse (Zhou et al., 2021). Due to vanilla attention's high computational demands, especially in GPU memory usage, we have scaled down the resolution to 128×128 for practical evaluation, instead of using the original 1024×1024 resolution of the turbulence data. FLOPs are calculated per sample. We have observed that while the FAVOR method from the Performer and the ProbSparse self-attention from the Informer significantly reduce computational complexity, adapting these sequence-to-sequence models for computer vision tasks including super-resolution necessitates further optimization, such as adopting patch-based learning similar to ViT (Dosovitskiy et al., 2021). This adaptation process may influence the computational cost metrics, potentially biasing them. For this reason, the computational cost metrics measured by the wall clock time may be biased due to our naive re-implementation of Performer and Informer not being optimized for this specific context. Despite these considerations, our findings indicate superior performance from the Galerkin-type self-attention. This is likely due to its specific design for operator learning problems. However, recent studies have reported variants of Galerkin-type attention that achieve even better results, which merits further investigation (Hao et al., 2023).

Table 4. Ablation study for self-attention mechanisms.

Model	Params. (M)	FLOPs (G)	Time (sec)	MSE
Vanilla	1.441	28.062	5.2147	3.382e-5
FAVOR	1.665	5.622	2.4746	1.002e-5
ProbSparse	1.681	5.644	2.4688	9.941e-6
Galerkin	1.709	2.708	0.3278	5.201e-6

Enhancement via Loss Prior. Table. 5 quantitatively summarizes the model performance, as measured by MSE, both with and without the inclusion of the proposed loss function. The experiments have been conducted with the upsampling ratio set at 4. The ablation study demonstrates that the inclusion of the proposed loss function enhances performance by at least 1.32% across various datasets.

Table 5. Ablation study for learnable frequency-aware loss prior.

Model design	Turbulence	Temperature	Kinetic Energy	Water Vapor
loss prior (-)	6.194e-6	2.061e-5	1.455e-4	3.169e-5
loss prior (+)	6.112e-6	1.983e-5	1.138e-4	3.112e-5
improvement	1.32%	3.78%	21.78%	1.79%

Furthermore, we investigate the correlation between high-frequency signals missing in deep learning predictions and the structure prior. We begin by calculating the error as the absolute difference between the deep learning predictions and the target. Subsequently, Pearson's correlation coefficient is utilized to assess the relationship between this error and the structure priors, each associated with distinct hyper-parameters. In our specific case, we observe that settings of $\alpha = 1$ and $\beta = 0.1$ yield the most effective structure prior, demonstrating a significant correlation (0.7978) with the error (See Fig.7). This insight allows us to employ this matrix to direct the network towards improved learning of high-frequency signals.

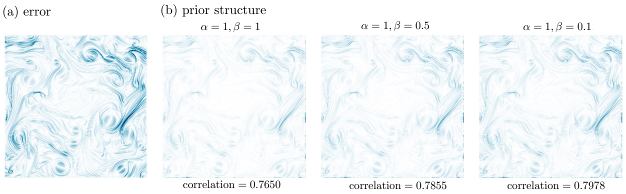

Figure 7. Illustration of the correlation between the high-frequency signals absent in deep learning predictions and the structure priors.

5. Conclusion

In this paper, we introduce the **Hi**erarchical **N**eural **O**perator **T**ransform**E**r (HiNOTE) for arbitrary-scale super-resolution. HiNOTE conceptualizes digital data as continuous functions, learning mappings between finite-dimensional function spaces, which enables training and generalization across various discretization levels. The process begins by transforming discretized low-resolution input into enhanced spatial-resolution feature maps. Subsequently, these maps are elevated to a higher-dimensional feature space by increasing the channel size to reveal latent features omitted in the original representation. This is followed by applying a hierarchical and self-attention-based kernel approximation before the final output mapping. HiNOTE demonstrates consistent state-of-the-art performance on both turbulence and weather data benchmarks, benefiting from its innovative architecture.

Acknowledgement

This research were supported by funding from the Advanced Scientific Computing Research program in the Department of Energy's Office of Science under project $B\&R\#KJ0402010$.

Impact Statement

Broader impacts The proposed model carries significant broader impacts across various domains, enhancing our ability to interpret and utilize data at unprecedented levels of detail. For instance, in environmental science, it can enhance satellite imagery resolution, enabling more precise climate models and improved decision-making for natural disaster response. In healthcare, it transforms medical imaging by offering detailed visualization of small anatomical features, which could facilitate earlier disease detection. Ultimately, the development of arbitrary-scale super-resolution methods promises to significantly impact society by improving our understanding and management of complex systems, from global ecosystems to human health.

Fairness and ethic issues Our research is committed to upholding ethical standards, focusing on developing models that ensure fairness, reduce biases, and protect privacy. We emphasize transparency in our methods and are willing to share our results to encourage ethical evaluation and peer review.

References

Akiva, P., Purri, M., and Leotta, M. Self-supervised material and texture representation learning for remote sensing tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8203–8215, 2022.

Arakawa, A. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. part i. *Journal of computational physics*, 135(2):103–114, 1997.

Brigham, E. O. *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1988.

Cao, J., Wang, Q., Xian, Y., Li, Y., Ni, B., Pi, Z., Zhang, K., Zhang, Y., Timofte, R., and Van Gool, L. Ciaosr: Continuous implicit attention-in-attention network for arbitrary-scale image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1796–1807, 2023.

Cao, S. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34: 24924–24940, 2021.

Chang, H., Yeung, D.-Y., and Xiong, Y. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pp. I–I. IEEE, 2004.

Chen, Y., Liu, S., and Wang, X. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8628–8638, 2021.

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2020.

Dong, C., Loy, C. C., He, K., and Tang, X. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pp. 184–199. Springer, 2014.

Dong, C., Loy, C. C., He, K., and Tang, X. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Dumoulin, V. and Visin, F. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

Fukami, K., Fukagata, K., and Taira, K. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019.

Gao, Z., Shi, X., Wang, H., Zhu, Y., Wang, Y. B., Li, M., and Yeung, D.-Y. Earthformer: Exploring space-time transformers for earth system forecasting. *Advances in Neural Information Processing Systems*, 35:25390–25403, 2022.

Gou, Y., Hu, P., Lv, J., Zhu, H., and Peng, X. Rethinking image super resolution from long-tailed distribution learning perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14327–14336, 2023.

Hao, Z., Wang, Z., Su, H., Ying, C., Dong, Y., Liu, S., Cheng, Z., Song, J., and Zhu, J. Gnot: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, pp. 12556–12569. PMLR, 2023.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730): 1999–2049, 2020.

Holmes, P. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.

Hu, X., Mu, H., Zhang, X., Wang, Z., Tan, T., and Sun, J. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1575–1584, 2019.

Jalal, A., Arvinte, M., Daras, G., Price, E., Dimakis, A. G., and Tamir, J. Robust compressed sensing mri with deep generative priors. *Advances in Neural Information Processing Systems*, 34:14938–14954, 2021.

Jiang, L., Dai, B., Wu, W., and Loy, C. C. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13919–13929, 2021.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.

Keys, R. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 1981.

Kim, J., Lee, J. K., and Lee, K. M. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1646–1654, 2016.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

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.

Lee, J. and Jin, K. H. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1929–1938, 2022.

Li, G., Lv, J., Tian, Y., Dou, Q., Wang, C., Xu, C., and Qin, J. Transformer-empowered multi-scale contextual matching and aggregation for multi-contrast mri super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20636–20645, 2022.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Li, Z., Shu, D., and Farimani, A. B. Scalable transformer for PDE surrogate modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=djyn8Q0anK.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., and Timofte, R. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1833–1844, 2021.

Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017.

Liu, H., Li, Z., Shang, F., Liu, Y., Wan, L., Feng, W., and Timofte, R. Arbitrary-scale super-resolution via deep learning: A comprehensive survey. *Information Fusion*, pp. 102015, 2023.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Luo, X., Qian, X., Urban, N., and Yoon, B.-J. Reinstating continuous climate patterns from small and discretized data. In *1st Workshop on the Synergy of Scientific and Machine Learning Modeling@ ICML2023*, 2023.

Pawar, S., San, O., Rasheed, A., and Vedula, P. Frame invariant neural network closures for kraichnan turbulence. *Physica A: Statistical Mechanics and its Applications*, 609:128327, 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.

Raonic, B., Molinaro, R., Ryck, T. D., Rohner, T., Bartolucci, F., Alaifari, R., Mishra, S., and de Bezenac, E. Convolutional neural operators for robust and accurate learning of PDEs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=MtekhXRP4h.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat, f. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.

Ren, P., Erichson, N. B., Subramanian, S., San, O., Lukic, Z., and Mahoney, M. W. Superbench: A super-resolution benchmark dataset for scientific machine learning. *arXiv preprint arXiv:2306.14070*, 2023.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

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.

Song, G., Sun, Q., Zhang, L., Su, R., Shi, J., and He, Y. Ope-sr: Orthogonal position encoding for designing a parameter-free upsampling module in arbitrary-scale image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10009–10020, 2023.

Stengel, K., Glaws, A., Hettinger, D., and King, R. N. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117(29):16805–16815, 2020.

Timofte, R., De Smet, V., and Van Gool, L. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE international conference on computer vision*, pp. 1920–1927, 2013.

Tran, A., Mathews, A., Xie, L., and Ong, C. S. Factorized fourier neural operators. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tmIiMPl4IPa.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Veillette, M., Samsi, S., and Mattioli, C. Sevir: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *Advances in Neural Information Processing Systems*, 33:22009–22019, 2020.

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.

Wei, M. and Zhang, X. Super-resolution neural operator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18247–18256, 2023.

Xie, Y., Franz, E., Chu, M., and Thuerey, N. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4): 1–15, 2018.

Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., and Sridhar, S. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pp. 641–676. Wiley Online Library, 2022.

Yang, J., Wright, J., Huang, T. S., and Ma, Y. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

Yang, Q., Harder, P., Ramesh, V., Hernandez-Garcia, A., Szwarcman, D., Sattigeri, P., Watson, C. D., and Rolnick, D. Fourier neural operators for arbitrary resolution climate data downscaling. In *ICLR 2023 Workshop on Tackling Climate Change with Machine Learning*, 2023. URL https://www.climatechange.ai/papers/iclr2023/59.

Yao, J.-E., Tsao, L.-Y., Lo, Y.-C., Tseng, R., Chang, C.-C., and Lee, C.-Y. Local implicit normalizing flow for arbitrary-scale image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1776–1785, 2023.

Yu, H., Huang, J., Zhao, F., Gu, J., Loy, C. C., Meng, D., Li, C., et al. Deep fourier up-sampling. *Advances in Neural Information Processing Systems*, 35:22995–23008, 2022.

Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., and Huang, T. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2472–2481, 2018.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

A. Data

A.1. Turbulent Flow

Turbulent flow is a fluid motion marked by irregular fluctuations in velocity and pressure. In this type of flow, fluid particles move chaotically, causing rapid changes in velocity and direction. The understanding of turbulent flow relies on the Navier-Stokes (NS) equations, which serve as a fundamental framework for studying fluid dynamics (Holmes, 2012). However, solving these equations becomes challenging when turbulence is present, as they couple the velocity field to pressure gradients:

$$\nabla \cdot \mathbf{u} = 0, \quad \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla \mathbf{p} + \nu \nabla^2 \mathbf{u} \tag{11}$$

The variables $\mathbf{u}$ and $\mathbf{p}$ represent the velocity field and pressure, respectively. ρ and ν stand for density and viscosity. In this study, we focus on two-dimensional Kraichnan turbulence in a doubly periodic square domain within $[0, 2\pi]^2$. The spatial domain is discretized using 2048^2 degrees of freedom. Solution variables of the NS equations are obtained through direct numerical simulation. A second-order energy-conserving Arakawa scheme computes the nonlinear Jacobian (Arakawa, 1997), and a second-order finite-difference scheme is employed for the Laplacian of the vorticity (Ren et al., 2023).

A.2. Global Weather Pattern

Global weather patterns depict the dominant atmospheric conditions and circulation features defining the Earth's climate globally. These patterns arise from interactions among elements of the Earth's atmosphere, including air temperature, pressure, humidity, and wind. These interactions span spatial and temporal scales over $\mathcal{O}(10)$ orders of magnitude, ranging from micrometers to planetary scales. In this study, we employ ERA5, an advanced atmospheric reanalysis dataset (Hersbach et al., 2020). Specifically, ERA5 has global coverage, ranging from the surface to the stratosphere, with a spatial resolution of 0.25 degrees (approximately 25 kilometers). When represented on a cartesian grid, these variables form a 720×1440 pixel field at any given altitude. Vertically, it is resolved into 37 pressure levels, offering detailed insights up to about 100 kilometers in altitude. The dataset provides hourly estimates of various atmospheric variables, facilitating detailed analyses of short-term weather events. Spanning from 1979 to near-real-time, ERA5 delivers a continuous, long-term record of Earth's atmospheric state. Its creation involves assimilating observations from diverse sources, such as satellite data, ground-based measurements, and meteorological observations, using an advanced numerical model. In this work, we employed three key variables: (1) kinetic energy at 10 meters above the surface, (2) surface temperature at 2 meters, and (3) total column water vapor. The variables are sampled daily at a frequency of 24 hours, spanning a period of 7 years.

A.3. SEVIR

The Storm EVent ImagRy (SEVIR) (Veillette et al., 2020) dataset is specifically curated to facilitate the development and evaluation of machine learning models in meteorology, with a particular emphasis on nowcasting severe weather events. SEVIR offers a substantial and well-organized collection of labeled examples depicting a range of weather phenomena, including thunderstorms, convective systems, and other related events. Key features of the SEVIR dataset include: (1) High-resolution Imagery: SEVIR encompasses high-resolution spatial and temporal satellite and radar imagery, capturing the intricate dynamics of storm development and progression; (2) Multimodal Data: The dataset incorporates data from various sources, such as satellite imagery (both visible and infrared), radar data, lightning maps, and derived products like Vertically Integrated Liquid (VIL) maps. This multimodal integration enables a comprehensive understanding of storm structures; (3) Event-based Sampling: SEVIR adopts an event-based sampling approach, concentrating on specific storm events. This method provides a focused dataset for the analysis of severe weather, including images captured before, during, and after significant events, which facilitates the temporal analysis of storm evolution; and (4) Wide Coverage: The dataset spans a broad geographic area, primarily across the continental United States, which experiences a diverse array of severe weather events. This extensive coverage enhances the general applicability of models trained on this data.

A.4. Data Summary

Overall, Table. 6 summarizes the datasets utilized in our experiments. The spatial resolution of these datasets has been carefully selected to enable efficient training without requiring multi-GPU computing resources.

Table 6. Summary of experiment benchmarks. Turbulence data is simulated using a time step of $\Delta t = 5 \times 10^{-4}$, while weather data is gathered at a sampling frequency of 24 hours.

Datasets	Spatial	Temporal	Train/Valid/Test
Turbulence	1024×1024	$0 \rightarrow 4$	700/200/100
Weather	720×1440	$2007 \rightarrow 2012$	700/200/100
SEVIR	768×768	2018	700/200/100
MRI	512×512	N/A	700/200/100

B. Model

B.1. Baselines

In this section, we provide additional training details for all baseline models. To ensure the reliability and consistency of our findings, each experiment was thoughtfully replicated five times. The models were implemented in PyTorch and experiments were conducted on an A100 GPU with 48GB.

- **SRCNN**: Dong et al. (2015) pioneered the use of a fully convolutional neural network for image super-resolution (SR), enabling end-to-end learning of the LR-to-HR mapping with minimal preprocessing. In our study, we employed SRCNN as a baseline for comparison and followed its default network design. To enhance training, we replaced the original Stochastic Gradient Descent (SGD) optimizer with ADAM (Kingma & Ba, 2015), balancing convergence speed and stability with a learning rate of 1×10^{-3}. Regularization was implemented using a weight decay factor of 1×10^{-5} to prevent overfitting and encourage generalization. Our training regimen extended over 200 epochs, with a batch size of 32 chosen for computational efficiency.

- **ESPCN**: Shi et al. (2016) introduce an innovative resolution enhancement approach through pixel-shuffle, facilitating deep neural network training within the low-resolution latent space. The study employs the default network architecture, with a learning rate of 1×10^{-3}, a batch size of 32, and a weight decay of 1×10^{-4}. Training spans 200 epochs, employing the Adam optimizer (Kingma & Ba, 2015).

- **EDSR**: Utilizing a deep residual network architecture with an extensive array of residual blocks, EDSR (Lim et al., 2017) effectively acquires the LR-to-HR image mapping while capturing hierarchical features. Our comparison study adheres to EDSR's default network configuration, employing 16 residual blocks with a hidden channel size of 64. For optimization, we set the learning rate to 1×10^{-4} and incorporate a weight decay of 1×10^{-5}. The batch size is fixed at 64, and training spans 300 epochs, facilitated by the ADAM optimizer (Kingma & Ba, 2015).

- **WDSR**: Yu et al. (2018) introduced WDSR to enhance reconstruction accuracy and computational efficiency by considering wider features before ReLU in residual blocks. They presented two architectures, WDSR-A and WDSR-B, with WDSR-B being deeper and more powerful but demanding greater computational resources. Our implementation employs 18 lightweight residual blocks with wide activation and a hidden channel of 32. Training utilizes a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5} over 300 epochs with the ADAM optimizer (Kingma & Ba, 2015). The batch size is set to 32.

- **SwinIR**: SwinIR (Liang et al., 2021) is built upon the sophisticated Swin Transformer architecture (Liu et al., 2021), leveraging its capabilities for local attention and cross-window interaction. Key architectural parameters include the use of 6 residual Swin Transformer blocks (RSTB), 6 Swin Transformer layers (STL), a window size of 8, a channel number of 180, and 6 attention heads. For training, a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5} are selected. The batch size for the training process is configured as 32.

- **MetaSR**: The Meta-Upscale Module (Hu et al., 2019) is composed of a stack of 16 residual blocks. In the encoding phase, we aim to extract 64 features. During the training phase, we have chosen a learning rate of 1×10^{-4} along with a weight decay of 1×10^{-5}. The training process is conducted with a batch size of 32.

- **LIIF**: Following the original approach (Chen et al., 2021), we utilize patches as inputs for the encoder. Denoting the batch size as B, we start by randomly selecting B scaling factors ($r_{1\sim B}$) from a uniform distribution $\mathcal{U}[1, 4]$. Then, we extract B patches from training images, each sized at $\{32r_i \times 32r_i\}_i^B$, while their down-sampled counterparts remain

32×32. For the ground-truth data, we convert these images into pixel samples, with 1024 samples sampled from each image to ensure consistent shapes within a batch. The encoder, denoted as $E(\cdot)$, is based on EDSR-baseline but excludes its up-sampling modules, generating a feature map of the same size as the input image. The decoding function, denoted as $f(\cdot)$, is implemented as a 5-layer MLP with ReLU activation functions and hidden dimensions of 256.

- **LTE**: Lee (Lee & Jin, 2022) introduced the Local Texture Estimator (LTE), a dominant-frequency estimator designed for natural images. This estimator allows an implicit function to capture fine details during the continuous reconstruction of images. When integrated with a deep super-resolution (SR) architecture, the LTE effectively characterizes image textures in 2D Fourier space. We configure LTE analogously to LIIF. The amplitude and frequency estimators are constructed using 3×3 convolutional layers, each with 256 output channels. This configuration is equivalent to a fully connected layer when the feature maps are unfolded. The phase estimator consists of a single fully connected layer with a hidden dimension of 128.

- **DFNO**: Consistent with the original paper (Yang et al., 2023), our implementation follows a specific architectural design: (1) Encoder: Modeled as a residual convolutional network inspired by super-resolution GAN generators (Wang et al., 2018); (2) Decoder: Implemented as a Fourier Neural Operator; and (3) Upsampling: Achieved using bicubic interpolation. Specifically, the encoder consists of five residual blocks, and the Fourier neural operator comprises four layers of Fourier integral operators with ReLU activation and batch normalization. We built and trained the DFNO model on a turbulence and weather dataset with a $\times 4$ upsampling factor (32×32 to 128×128). Subsequently, we evaluated its performance using various upsampling factors.

- **SRNO**: The SRNO network (Wei & Zhang, 2023) architecture features a key component, namely a feature encoder denoted as $E(\cdot)$. To construct this encoder, we adopt the EDSR-baseline architecture and exclude its upsampling layers, while maintaining output channel dimensions at $d_e = 64$. We also include a multi-head attention mechanism, which involves dividing queries, keys, and values into n_h segments, each with a dimension of d_z/n_h. In our specific implementation, we set the embedding dimension as $d_z = 256$ and the number of heads as $n_h = 16$, resulting in 16-dimensional output values from the attention mechanism. The kernel integral operator is applied twice.

B.2. Our Approach

B.2.1. Improvements

- **Feature Refining** In contemporary super-resolution methods, the upsampling layer is typically positioned towards the end of the network. We suggest that enhancing feature maps at the beginning of the network is more effective for SR tasks.

- **Neural Operator** Implicit neural representations, often parameterized by multi-layer perceptrons (MLPs), struggle with high-frequency information learning due to their point-wise spatial behavior, a phenomenon known as spectral bias. To address this, we substitute the MLP-based inference network with a neural operator. This operator treats images as continuous functions instead of 2D pixel arrays, where each image is a discretization of an underlying function.

B.2.2. Loss function

We propose two approaches to incorporate the loss prior into training our HiNOTE model: (1) utilizing a focal loss-based technique, and (2) employing a two-stage network training process.

Approach 1 Similar to the concept of focal loss (Jiang et al., 2021; Gou et al., 2023), the loss function can be formulated as

$$\mathcal{L} = \frac{1}{N}\sum_{j=1}^{N}\mathcal{L}_j \quad \text{where} \quad \mathcal{L}_j = \mathbf{W}(\boldsymbol{p}^{(j)};\alpha_p,\beta_p)\times\mathbf{W}(\hat{\boldsymbol{b}}^{(j)};\alpha_{\hat{\boldsymbol{b}}},\beta_{\hat{\boldsymbol{b}}})\times\left|\mathcal{G}_\theta\left(\boldsymbol{a}^{(j)}\right)+\mathcal{R}\left(\boldsymbol{a}^{(j)}\right)-\boldsymbol{b}^{(j)}\right| \tag{12}$$

In Eq. 12, the prior loss $\mathbf{W}(\boldsymbol{p}^{(j)};\alpha_p,\beta_p)$ quantifies the discrepancy between the target and the output of spectral resizing, while $\mathbf{W}(\hat{\boldsymbol{b}}^{(j)};\alpha_{\hat{\boldsymbol{b}}},\beta_{\hat{\boldsymbol{b}}})$ assesses the difference between the target and the deep learning prediction.

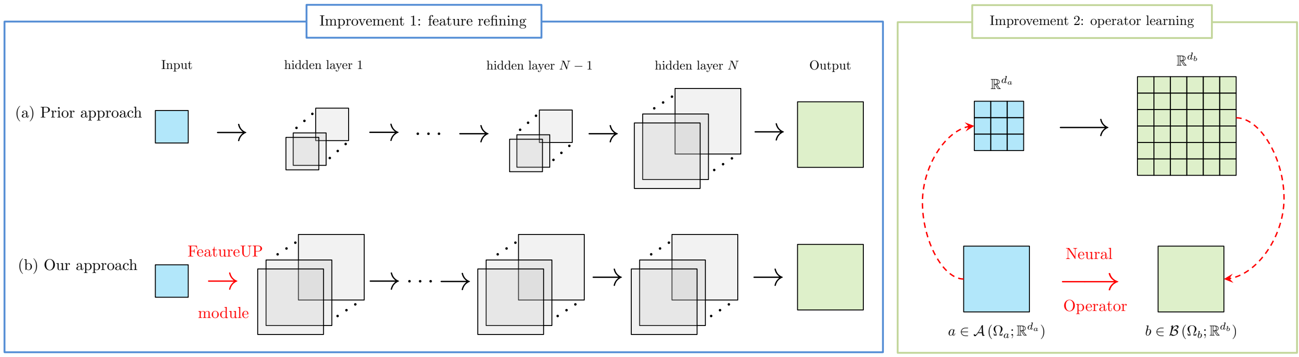

Figure 8. Comparison of the proposed model's enhancements over the previous approach.

Approach 2 In the first stage, we employ the standard training procedure. The problem of learning an operator that approximates the mapping between $\mathcal{A}$ and $\mathcal{B}$ naturally takes the form of an empirical risk minimization problem

$$\theta^{\dagger} = \arg\min_{\theta \in \Theta} \mathbb{E}_{\mathbf{a}} \left[\mathcal{C}\left(\mathcal{G}(\mathbf{a}, \theta), \mathcal{G}^{\dagger}(\mathbf{a}) \right) \right] \approx \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{j=1}^{N} \left\| \boldsymbol{b}^{(j)} - \mathcal{G}_{\theta}\left(\boldsymbol{a}^{(j)} \right) \right\|_{\mathcal{B}}^{2} \tag{13}$$

where function $\mathcal{C} : \mathcal{B} \times \mathcal{B} \to \mathbb{R}$ serves as a cost functional, quantifying the distance within the space $\mathcal{B}$. The optimized parameter values from stage 1 serve as the initial values for stage 2, aiming to solve

$$\arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{j=1}^{N} \mathbf{W}(\boldsymbol{p}^{(j)}; \alpha, \beta) \cdot \left\| \boldsymbol{b}^{(j)} - \mathcal{G}_{\theta}\left(\boldsymbol{a}^{(j)} \right) \right\|_{\mathcal{B}}^{2} \tag{14}$$

In stage 2, the model prioritizes learning from challenging-to-fit pixels during the initial fitting stage.

B.2.3. Hyperparameters

In deep learning, the design of network architectures and the optimization of hyperparameters are substantial challenges, typically addressed through empirical, problem-specific methods. To facilitate a fair comparison, we have executed extensive hyperparameter tuning and architecture searches for each model under consideration. The specific model configurations for our proposed method are outlined in Table 7.

Table 7. Model configurations, hyperparameters, and associated ranges.

Hyperparameters	Values
Upsample ratio	$\{\times 1, \times 2, \times 4\}$
Nonlinear activation	{ ReLU, LeakyReLU, ELU, SELU, GELU, RReLU }
Attention width	$\{128, 160, 192, 224, 256\}$
Attention head	$\{2, 4, 8, 16, 32\}$

B.3. Additional qualitative results

We present the test predictions of HiNOTE alongside various *arbitrary-scale* super-resolution baseline methods. For each figure, the super-resolved predictions are depicted in the top row, while the bottom row features error maps with respect to the reference data, with darker pixels indicating greater errors. This visual comparison not only highlights the accuracy of our model in generating high-resolution predictions but also provides a quantitative assessment of its performance by visualizing the error distribution across the domain.

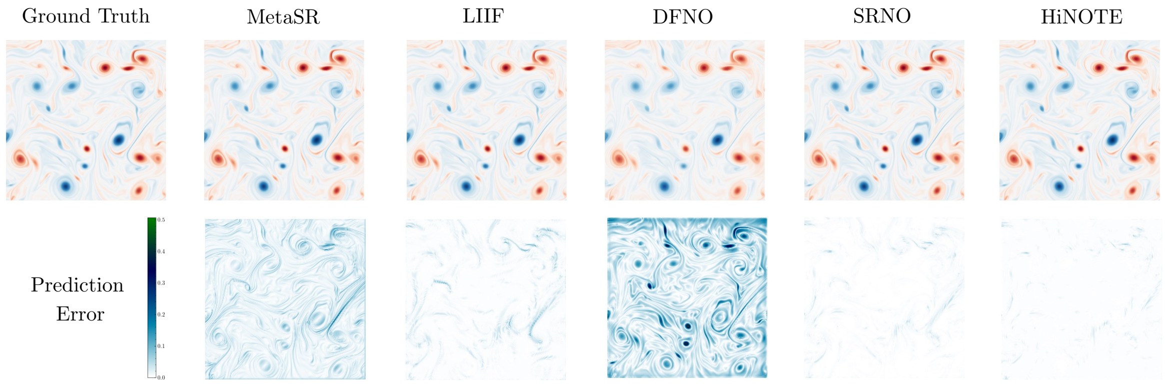

Figure 9. Qualitative assessment of the models on turbulence flow data.

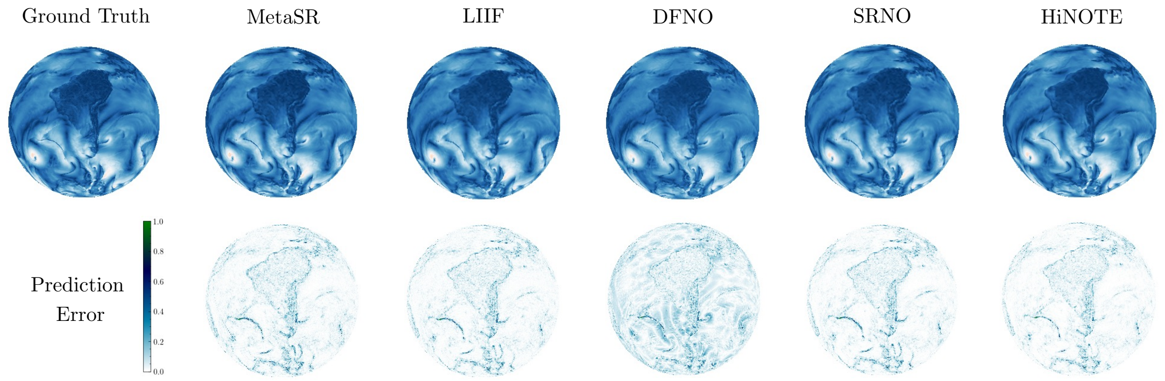

Figure 10. Qualitative assessment of the models on kinetic energy data.

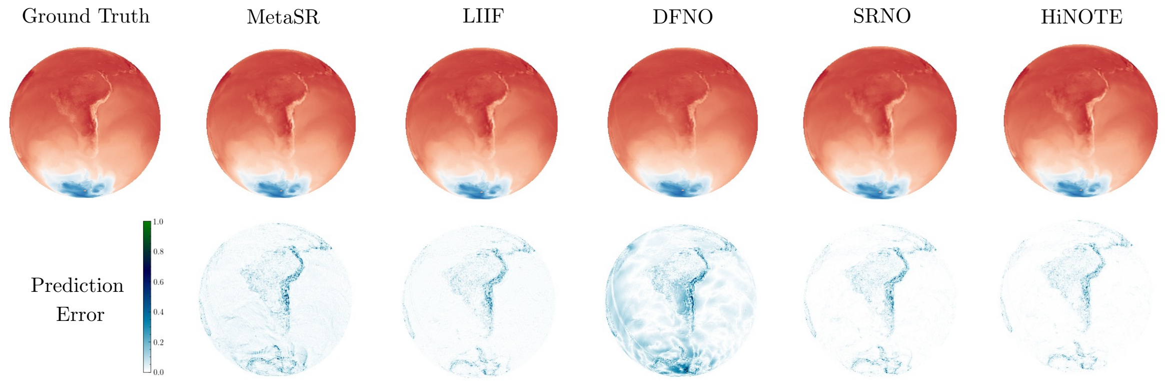

Figure 11. Qualitative assessment of the models on temperature data.

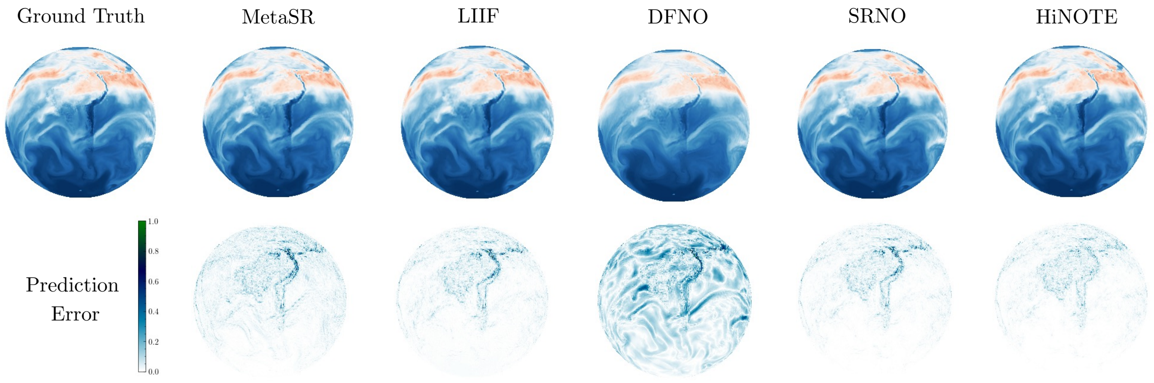

Figure 12. Qualitative assessment of the models on water vapor data.

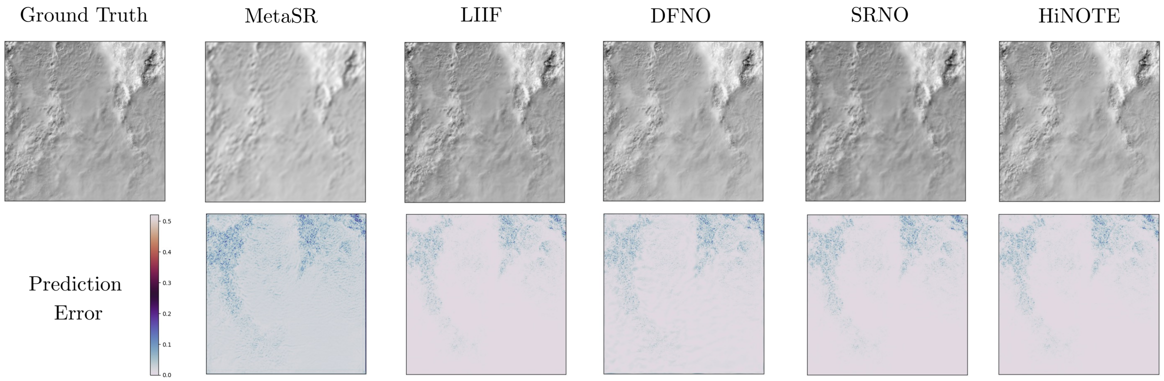

Figure 13. Qualitative assessment of the models on SEVIR data.

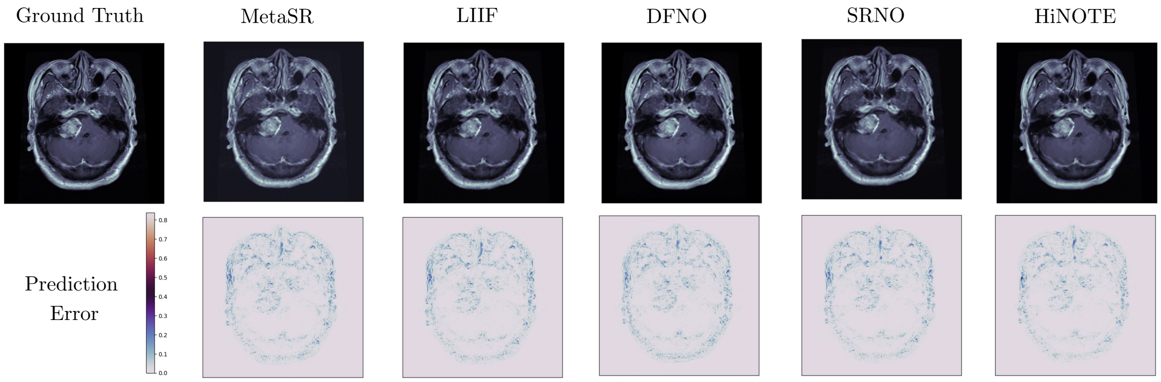

Figure 14. Qualitative assessment of the models on magnetic resonance imaging data.

B.4. Additional quantitative results

Here, we present comprehensive quantitative results for the arbitrary-scale super-resolution models applied to the SEVIR and MRI datasets.

Table 8. Quantitative comparison results for the *arbitrary-scale* SR tasks on the SEVIR data.

	×4.6			×8.2			×15.7			×32		
Model	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
MetaSR	5.630e-4	32.378	0.865	4.475e-3	23.376	0.675	9.420e-3	20.143	0.655	1.362e-2	18.542	0.685
LIIF	1.099e-4	39.473	0.965	4.059e-4	33.799	0.923	8.763e-4	30.457	0.894	1.504e-3	28.110	0.879
DFNO	1.729e-4	37.504	0.957	4.606e-4	33.251	0.915	9.226e-4	30.234	0.895	1.423e-3	28.349	0.881
SRNO	1.093e-4	39.488	0.964	3.312e-4	34.674	0.928	6.730e-4	31.595	0.905	1.119e-3	29.385	0.893
HiNOTE	1.048e-4	39.680	0.969	3.191e-4	34.845	0.931	6.302e-4	31.889	0.911	9.981e-4	29.892	0.902

Table 9. Quantitative comparison results for the *arbitrary-scale* SR tasks on the SEVIR data.

	×4.6			×8.2			×15.7			×32		
Model	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
MetaSR	4.761e-4	33.222	0.942	3.499e-3	24.559	0.784	9.841e-3	20.069	0.680	1.853e-2	17.320	0.624
LIIF	6.348e-4	31.973	0.916	2.327e-3	26.331	0.815	5.773e-3	22.385	0.734	1.123e-2	19.495	0.676
DFNO	4.937e-4	33.065	0.928	2.284e-3	26.412	0.830	6.161e-3	22.103	0.754	1.166e-2	19.330	0.663
SRNO	4.464e-4	33.502	0.937	2.125e-3	26.726	0.839	5.573e-3	22.538	0.759	1.176e-2	19.293	0.681
HiNOTE	3.805e-4	34.195	0.947	1.808e-3	27.427	0.871	4.806e-3	23.181	0.786	1.076e-2	19.678	0.704