Latent Spectral Fourier Neural Operators

1st Kaiyao Tan
School of Computer Science and Engineering
Sun Yat-sen University
Guangzhou, China
tanky5@mail2.sysu.edu.cn

2nd Qingsong Zou*
School of Computer Science and Engineering
Guangdong Province Key Laboratory of Computational Science
Sun Yat-sen University
Guangzhou, China
mcszqs@mail.sysu.edu.cn

***Abstract*—Fourier Neural Operators (FNOs) are widely used for PDE surrogate modeling, but their spectral kernels are learned separately for retained Fourier modes. As spectral resolution increases, this parameterization increases model size and often requires frequency truncation, which can hurt high-frequency accuracy. We propose the Latent Spectral Fourier Neural Operator (LS-FNO), which keeps the standard FFT–multiply–IFFT structure of FNOs but replaces discrete spectral weights with a continuous generator conditioned on frequency coordinates. Instead of learning each Fourier mode independently, LS-FNO represents spectral kernels through a compact low-rank latent space, where a shared encoder maps frequencies to latent codes and a decoder reconstructs complex-valued kernels. We implement the encoder with a KAN-based architecture to model fine spectral variation more effectively. Across six PDE benchmarks, LS-FNO achieves lower errors than strong baselines, with the largest gains on high-frequency-dominated tasks. On 1D Compressible CFD, it reduces relative error by 42%, while using up to 77% fewer parameters than enlarged FNO variants.**

***Index Terms*—Fourier Neural Operators, Scientific Machine Learning, Kolmogorov-Arnold Networks, Latent Representation, Operator Learning.**

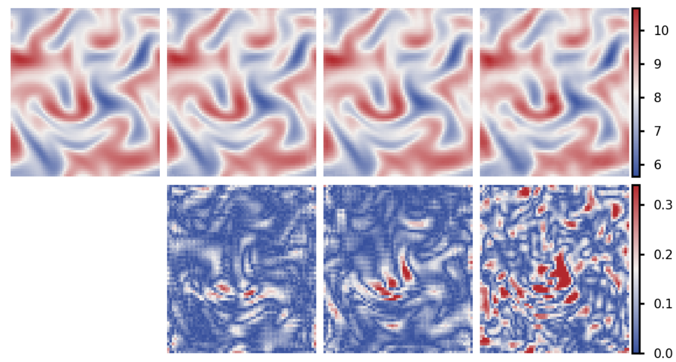

Fig. 1. **Qualitative comparison on the 2D Compressible CFD benchmark.** The top row visualizes the ground-truth target field (left) and the predictions produced by **LS-FNO**, **AM-FNO**, and **FNO** (from left to right), all rendered with a shared colormap and an identical value range (top colorbar). The bottom row reports the corresponding *absolute error maps* $|\hat{u} - u|$ for each model, using a shared colormap and a common error scale (bottom colorbar). The bottom-left panel is intentionally left blank for layout consistency.

I. Introduction

Learning surrogate models for partial differential equations (PDEs) has become a central topic in scientific machine learning (SciML) [1, 2], enabling fast and accurate approximations of complex physical systems ranging from computational fluid dynamics (CFD) to large-scale weather forecasting [3, 4]. Unlike classical numerical solvers or pointwise regression models [5, 6], *operator learning* seeks to approximate mappings between infinite-dimensional function spaces, allowing models to generalize across resolutions and input discretizations [7, 8].n This paradigm has inspired a wide range of neural architectures, including DeepONet [7], graph-based operators [9, 10], and spectral methods based on wavelets or Clifford algebras [11, 12]. Among these approaches, Fourier Neural Operators (FNOs) [4] have emerged as a widely adopted backbone due to their efficient global convolution via Fast Fourier Transforms (FFT) and theoretical universality guarantees [13].

Despite their empirical success, standard FNOs suffer from a fundamental limitation in the *parameterization of spectral kernels*. In practice, the Fourier-domain kernel $\boldsymbol{R}(\boldsymbol{k})$ is represented as a discrete, mode-wise tensor with independent parameters for each retained frequency. As a result, the parameter complexity scales linearly with the spectral resolution [14, 15], making high-resolution modeling prohibitively expensive. To control memory and computation, FNOs typically rely on aggressive frequency truncation, which discards high-frequency components that are essential for capturing shocks, discontinuities, and fine-scale turbulent structures [16, 17].

Recent work has explored several directions to alleviate this bottleneck, including low-rank tensor factorizations [14, 18] and attention-based or Transformer-style operators [19]–[21]. More closely related to our work are *amortized spectral methods*, such as Adaptive FNO (AFNO) [22] and Amortized FNO (AM-FNO) [15], which generate spectral weights through coordinate-based neural networks. While these approaches reduce memory costs, they typically regress spectral kernels directly from frequency coordinates using standard MLPs. However, MLPs are known to suffer from *spectral bias* [23], preferentially learning low-frequency functions while struggling to converge on complex, high-frequency oscillatory patterns. Consequently, direct coordinate-based regression often yields overly smooth kernels that fail to preserve the high-frequency energy required for shock-dominated flows.

In this work, we propose the **Latent Spectral Fourier Neural Operator (LS-FNO)**, which rethinks spectral modeling as a problem of *continuous latent representation learning*. Instead of storing discrete mode-wise kernels or directly regressing weights via MLPs, we hypothesize that the spectral kernels of physical operators lie on a shared, low-dimensional **latent spectral manifold**. Under this perspective, the kernel $\boldsymbol{R}(\boldsymbol{k})$ at any frequency $\boldsymbol{k}$ is generated by decoding a compact latent representation, thereby enforcing global coherence across frequencies. Crucially, to realize this idea, we employ Kolmogorov–Arnold Networks (KANs) [24] as the continuous spectral encoder. Unlike MLPs which rely on global activation functions, KANs utilize learnable splines with **local support** on network edges. This architecture provides a strong inductive bias for capturing local variations in spectral density, effectively mitigating spectral bias and enabling accurate reconstruction of high-frequency spectral content.

The main contributions of this work are summarized as follows:

- **Latent spectral parameterization.** We introduce a generative formulation that decouples model size from spectral resolution by embedding spectral kernels into a shared low-dimensional latent manifold, eliminating the need for heuristic frequency truncation.
- **Spline-based spectral encoding.** We leverage KANs as continuous frequency-to-latent encoders. We demonstrate that the local support of spline activations provides a superior inductive bias compared to MLPs, significantly improving high-frequency accuracy.
- **Continuous spectral representation.** LS-FNO learns a resolution-independent mapping from frequency coordinates to kernel weights, providing a principled framework for physically consistent operator learning across arbitrary discretizations.

II. Related Work

Neural Operators and Architectures. Since the introduction of FNO [4], neural operators have evolved to handle complex geometries via mapping [25] or graph-based message passing [10, 21]. Transformer-based variants [19, 20] and physics-informed methods [26] have further improved expressivity. However, these models typically rely on discrete parameterizations that scale linearly with resolution, creating a bottleneck for high-dimensional problems [13].

Parameter Efficiency and Generative Modeling. To improve efficiency, recent works employ tensor factorization [14, 18] or multi-scale U-Net structures [16, 17]. A more scalable paradigm is *coordinate-based weight generation*, exemplified by AFNO [22] and AM-FNO [15], which generate spectral weights as functions of frequency coordinates. LS-FNO advances this direction by introducing a continuous Latent Spectral Manifold with an explicit low-rank bottleneck, promoting shared structure and global coherence across frequencies. Furthermore, LS-FNO leverages a KAN-based encoder [24] to parameterize this manifold, providing a spline-based inductive bias that mitigates spectral bias [23] and improves high-frequency expressivity compared to direct regression approaches.

III. Preliminaries

A. Operator Learning

Let $D \subset \mathbb{R}^d$ be a bounded domain. Operator learning aims to approximate a mapping $\mathcal{G} : \mathcal{A} \to \mathcal{U}$ between infinite-dimensional function spaces, where $\mathcal{A}$ typically denotes the space of input fields (e.g., coefficients, forcings, or initial conditions) and $\mathcal{U}$ denotes the solution space. Given a dataset of N pairs $\{(a_i, u_i)\}_{i=1}^N$ sampled on a spatial grid, a neural operator $\mathcal{G}_\theta$ is trained by minimizing an empirical loss, e.g., the relative L^2 error

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \frac{\|\mathcal{G}_\theta(a_i) - u_i\|_2}{\|u_i\|_2}. \tag{1}$$

A hallmark of neural operators is *discretization invariance*: the learned mapping is expected to generalize across discretizations and resolutions without retraining [8].

B. Fourier Neural Operators

Fourier Neural Operators (FNOs) realize global mixing via spectral convolution. Let $h^\ell(x) \in \mathbb{R}^{d_h}$ be the feature field at layer ℓ. A standard FNO layer updates

$$h^{\ell+1}(x) = \sigma\big(\boldsymbol{W} h^\ell(x) + \mathcal{F}^{-1}\big(\boldsymbol{R}(\boldsymbol{k}) \cdot \mathcal{F}(h^\ell)(\boldsymbol{k})\big)(x)\big), \tag{2}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, $\boldsymbol{W}$ is a pointwise linear map, and $\boldsymbol{R}(\boldsymbol{k}) \in \mathbb{C}^{d_h \times d_h}$ denotes the spectral kernel at frequency mode $\boldsymbol{k}$ [4].

In practice, FNO parameterizes $\boldsymbol{R}$ on a discrete frequency grid and stores the weights as a tensor, typically keeping only a truncated set of low-frequency modes to control memory and compute. This discrete, mode-wise parameterization leads to parameter growth with spectral resolution and may discard high-frequency information that is important for sharp transitions and fine-scale dynamics.

IV. Method

We propose the **Latent Spectral Fourier Neural Operator (LS-FNO)**, a generative spectral architecture that addresses the parameter scalability and high-frequency modeling limitations of standard FNOs. Importantly, LS-FNO *does not alter the operator form of FNO*: the global convolution is still implemented via FFT, mode-wise spectral multiplication, and inverse FFT. Instead, our method exclusively reformulates *how the spectral kernel is parameterized*.

Specifically, while standard FNOs store an independent complex-valued matrix for each frequency mode, LS-FNO treats the spectral kernel as a *continuous function over the frequency domain*. For any frequency coordinate $\boldsymbol{k}$, the corresponding kernel $\boldsymbol{R}(\boldsymbol{k})$ is synthesized on-the-fly by a shared neural generator. This design decouples the number of learnable parameters from the spectral resolution and enforces structural coherence across frequencies. As illustrated in Fig. 2, LS-FNO consists of (i) a standard spectral operator layer and (ii) a latent spectral generator that produces complex-valued kernel weights.

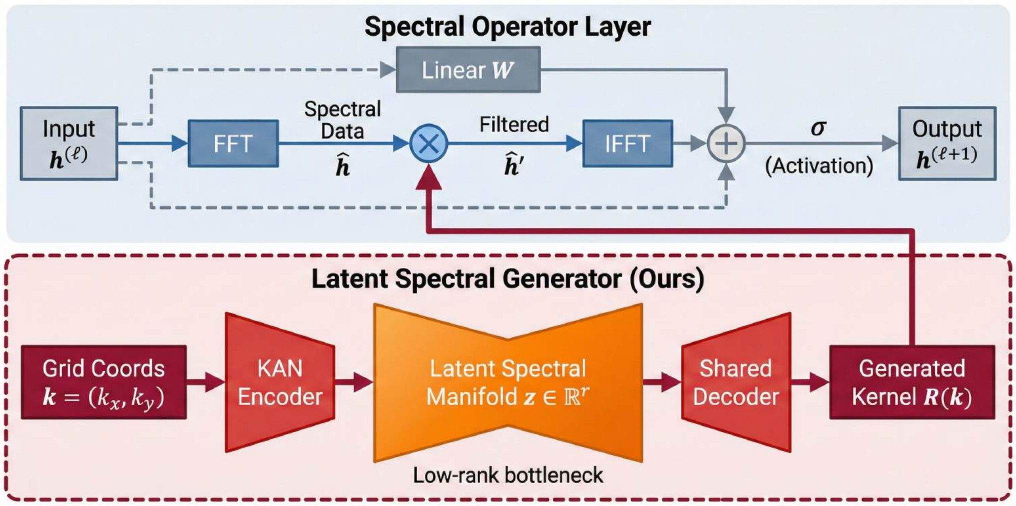

Fig. 2. **Architecture of the Latent Spectral Fourier Neural Operator (LS-FNO).** LS-FNO preserves the standard global convolution structure of Fourier Neural Operators via FFT, while replacing the discrete, mode-wise spectral weights with a frequency-conditioned generator. The generator maps frequency coordinates $\boldsymbol{k}$ to a low-dimensional latent code and decodes it into complex-valued spectral kernels $\boldsymbol{R}(\boldsymbol{k})$, enabling resolution-independent parameterization.

A. Spectral Operator Layer

We follow the standard Fourier Neural Operator update. Let $h^\ell(x) \in \mathbb{R}^{d_h}$ denote the feature field at layer ℓ. Denote by $\widehat{h}^\ell(\boldsymbol{k}) = \mathcal{F}(h^\ell)(\boldsymbol{k}) \in \mathbb{C}^{d_h}$ its Fourier coefficients. For each frequency mode $\boldsymbol{k}$, the spectral kernel $\boldsymbol{R}(\boldsymbol{k}) \in \mathbb{C}^{d_h \times d_h}$ acts as a *mode-wise linear map* on the channel dimension. The kernel integral operator is implemented in the Fourier domain as

$$(\mathcal{K}h^\ell)(x) = \mathcal{F}^{-1}\Big(\boldsymbol{R}(\boldsymbol{k})\,\widehat{h}^\ell(\boldsymbol{k})\Big)\,(x), \tag{3}$$

i.e., a mode-wise matrix–vector multiplication for each $\boldsymbol{k}$, followed by an inverse FFT. The layer update is then given by

$$h^{\ell+1}(x) = \sigma\big(\boldsymbol{W}h^\ell(x) + (\mathcal{K}h^\ell)(x)\big)\,, \tag{4}$$

where $\boldsymbol{W}$ is a pointwise linear map and σ denotes a nonlinear activation.

B. Latent Spectral Generator

We parameterize the spectral kernel using a frequency-conditioned generator $\mathcal{G}_\phi$:

$$\boldsymbol{R}(\boldsymbol{k}) = \mathcal{G}_\phi(\boldsymbol{k}). \tag{5}$$

To enforce parameter efficiency and global spectral coherence, we adopt a bottleneck-and-reconstruct design:

$$\boldsymbol{z}_{\boldsymbol{k}} = \mathcal{E}_\phi(\boldsymbol{k}) \in \mathbb{R}^r, \qquad \boldsymbol{R}(\boldsymbol{k}) = \mathcal{D}(\boldsymbol{z}_{\boldsymbol{k}}) \in \mathbb{C}^{d_h \times d_h}, \tag{6}$$

where $r \ll d_h^2$ is the latent rank. This low-dimensional bottleneck induces a shared latent spectral manifold across frequencies, encouraging smoothness and structural consistency while suppressing frequency-wise overfitting.

a) KAN-based Encoder: To implement the encoder $\mathcal{E}_\phi$, we employ KANs [24], which offer a spline-based alternative to standard MLPs. Unlike MLPs that rely on fixed activation functions at nodes, KANs parameterize learnable nonlinearities on the *edges* of the network. Specifically, for an input frequency coordinate $\boldsymbol{k} = (k_1, \dots, k_d)$, the mapping to the latent code $\boldsymbol{z}_{\boldsymbol{k}} \in \mathbb{R}^r$ is made up of layers where the computation corresponds to a sum of univariate functions. A single KAN layer mapping dimension d_{in} to d_{out} is defined as:

$$z_j = \sum_{i=1}^{d_{in}} \phi_{i,j}(k_i), \quad j = 1, \dots, d_{out}, \tag{7}$$

where each $\phi_{i,j}(\cdot)$ is a learnable univariate function parameterized as a B-spline:

$$\phi_{i,j}(x) = \sum_p c_{i,j,p} B_p(x) + w_{i,j}\sigma(x), \tag{8}$$

with $\{c_{i,j,p}\}$ being the trainable control coefficients, $B_p(x)$ being the basis B-spline, and the second term providing a residual connection for training stability. In our context, the input dimension corresponds to the spatial dimension ($d_{in} = d$) and the output dimension to the latent rank ($d_{out} = r$).

Crucially, the parameterization of the B-spline introduces a strong inductive bias of *local support*: modifying a coefficient c only affects the output of the function in a small local interval of the input frequency k_i. This property prevents the *spectral bias* typically observed in global MLP activations (e.g. ReLU or Tanh), allowing the encoder to accurately capture high-frequency spectral variations without overfitting or disrupting the learned low-frequency structures.

b) Complex-valued Decoding: The decoder $\mathcal{D}$ reconstructs the complex-valued kernel matrix from the latent

TABLE I
COMPARISON OF RELATIVE L^2 ERROR ACROSS 6 BENCHMARKS. BEST RESULTS ARE **BOLDED**, SECOND BEST ARE UNDERLINED. THE SYMBOL "-" DENOTES THAT THE MODEL IS INAPPLICABLE TO THE BENCHMARK OR RESULTS ARE UNREPORTED.

Model	NS-2D	Darcy	Pipe	Airfoil	CFD 1D	CFD 2D
FNO	1.56e-1	1.08e-2	-	-	2.93e-2	5.36e-3
AFNO	2.17e-1	3.17e-2	1.72e-2	9.88e-3	-	6.72e-2
Geo-FNO	1.56e-1	1.08e-2	6.70e-3	1.38e-2	2.93e-2	5.36e-3
OFormer	1.71e-1	1.24e-2	9.59e-3	1.83e-2	-	-
U-FNO	1.22e-1	1.24e-2	5.76e-3	1.05e-2	2.44e-2	4.52e-3
F-FNO	1.74e-1	9.92e-3	5.99e-3	1.00e-2	2.54e-2	7.86e-3
LSM	1.64e-1	7.01e-3	5.20e-3	6.39e-3	-	7.62e-2
AM-FNO	1.08e-1	4.28e-3	3.54e-3	6.06e-3	1.83e-2	2.70e-3
LS-FNO (Ours)	**9.98e-2**	**3.60e-3**	**3.42e-3**	**5.62e-3**	**1.07e-2**	**2.23e-3**

code. To maintain computational efficiency, we employ linear projections for the real and imaginary parts:

$$\boldsymbol{R}(\boldsymbol{k}) = (\boldsymbol{W}_{re}\boldsymbol{z}_{\boldsymbol{k}} + \boldsymbol{b}_{re}) + i\,(\boldsymbol{W}_{im}\boldsymbol{z}_{\boldsymbol{k}} + \boldsymbol{b}_{im}), \quad (9)$$

where $\boldsymbol{W}_{re}, \boldsymbol{W}_{im} \in \mathbb{R}^{d_h^2 \times r}$ are learnable weight matrices (reshaped to $d_h \times d_h \times r$). This linear decoding design preserves the algebraic structure of complex-valued spectral kernels while ensuring the generator remains lightweight.

C. Complexity Analysis

In a standard FNO layer, the spectral kernel stores an independent matrix for each retained mode, resulting in parameter complexity

$$\mathcal{O}(|\mathcal{K}| \cdot d_h^2),$$

where $|\mathcal{K}|$ is the number of Fourier modes (often after truncation). In LS-FNO, the kernel matrices are generated from a shared latent generator with a rank-r bottleneck, so the learnable parameters scale as

$$\mathcal{O}(N_{\text{spline}} + r \cdot d_h^2),$$

which is *independent* of $|\mathcal{K}|$. Therefore, LS-FNO decouples model size from spectral resolution and can exploit higher spectral resolutions without increasing the number of kernel parameters (at the cost of evaluating the generator per mode).

V. EXPERIMENTS

We conduct comprehensive experiments to evaluate the effectiveness of the proposed LS-FNO. Our evaluation focuses on three aspects: (i) **Predictive Accuracy**, via comparisons with state-of-the-art neural operators on standard PDE benchmarks; (ii) **Parameter Efficiency**, in terms of model size and training cost; (iii) **Design Validation**, through ablation studies that analyze the role of the latent spectral manifold and the KAN-based encoder.

A. Experimental Setup

1) Benchmarks: We evaluate LS-FNO on six canonical PDE benchmarks widely adopted in neural operator research [4, 15, 27], covering both incompressible and compressible flows across regular and irregular geometries. Specifically, we utilize the Darcy Flow and Navier-Stokes (NS-2D) datasets to assess performance on standard regular grids. To evaluate generalization capability on irregular domains, we employ the Pipe Flow and Airfoil benchmarks. Furthermore, we include the Compressible CFD (1D & 2D) problems [27] to test the model's ability to capture shocks and high-frequency features. A detailed summary of the dataset configurations is provided in Table II.

TABLE II
OVERVIEW OF BENCHMARK DATASETS

Benchmark	Spatial Dim	Resolution	N_{train}	N_{test}
Darcy Flow	2	85×85	1000	200
Navier-Stokes (2D)	2	64×64	1000	200
Pipe Flow	2	129×129	1000	200
Airfoil	2	221×51	1000	200
CFD-1D	1	1024	1000	200
CFD-2D	2	64×64	1000	200

2) Baselines: We compare LS-FNO against a comprehensive suite of state-of-the-art neural operators. As the foundational baseline, we include the standard FNO [4] with mode-wise parameterization. We also compare against its advanced variants: Geo-FNO [25] for irregular meshes, U-FNO [17] for multi-scale feature integration, F-FNO [14] for parameter efficiency, and the transformer-based AFNO [22]. Beyond FNO variants, we include OFormer [20] and LSM [28] as representative spectral baselines. Most importantly, we benchmark directly against AM-FNO [15], which utilizes amortized MLPs or KANs for weight generation, serving as the most direct competitor to our approach.

3) Implementation Details: All models are implemented in PyTorch and trained for 500 epochs using the AdamW optimizer [29] with a cosine annealing learning-rate scheduler [30]. The initial learning rate is set to 10^{-3} and the weight decay to 10^{-4}. All models consist of 4 operator layers with a uniform hidden width of 32 and process all available frequency modes without truncation. We use the Gaussian Error Linear Unit (GELU) [31] as the activation function.

For **LS-FNO**, the spectral generator employs a KAN-based encoder with a default grid size $G \in \{32, 48\}$, and the latent rank is set to $r = 16$ unless otherwise specified. These settings are fixed across all benchmarks, with further analysis provided in the ablation study. The training loss and evaluation metric are based on the relative L^2 error. All experiments are conducted on a single NVIDIA RTX 4090 GPU.

B. Main Results

Table I shows that LS-FNO achieves strong and consistent performance across all six benchmarks. In high-frequency regimes, which are particularly challenging for spectral methods, LS-FNO yields substantial error reductions, with relative improvements of up to 42% on CFD-1D and 17% on CFD-2D compared to the strongest baselines. On regular grids, LS-FNO outperforms recent amortized spectral methods on Darcy Flow and NS-2D, while on irregular geometries it consistently surpasses geometry-aware baselines on Pipe and Airfoil flows. Overall, these results indicate that the proposed continuous latent spectral formulation effectively captures fine-scale dynamics without relying on frequency truncation.

C. Ablation Study

TABLE III
ABLATION STUDY ON DARCY AND CFD-1D BENCHMARKS. WE REPORT RELATIVE L_2 ERRORS UNDER DIFFERENT ENCODER ARCHITECTURES, LATENT RANKS (r), KAN GRID SIZES (G), AND MODEL WIDTHS (W). BEST RESULTS ARE **BOLDED**.

Setting	Darcy	CFD-1D
1. Encoder architecture		
MLP encoder	1.56e-2	1.75e-2
KAN encoder	**3.72e-3**	**1.20e-2**
2. Latent rank (r)		
$r = 4$	4.04e-3	1.32e-2
$r = 8$	3.87e-3	1.26e-2
$r = 16$	**3.72e-3**	1.20e-2
$r = 32$	3.86e-3	1.21e-2
3. KAN grid size (G)		
$G = 16$	4.27e-3	1.36e-2
$G = 24$	3.99e-3	1.26e-2
$G = 32$	3.72e-3	1.20e-2
$G = 40$	**3.60e-3**	**1.16e-2**
$G = 48$	3.65e-3	1.21e-2
4. Model width (W)		
$W = 16$	3.99e-3	1.69e-2
$W = 24$	3.90e-3	1.40e-2
$W = 32$	**3.72e-3**	1.20e-2
$W = 40$	3.93e-3	**1.07e-2**
$W = 48$	3.75e-3	1.09e-2

We conduct controlled ablation studies to examine the impact of the spectral encoder and the low-rank latent representation. Quantitative results are reported in Table III, and the corresponding frequency-domain behavior is illustrated in Fig. 3.

a) Effect of the spectral encoder.: Replacing the MLP encoder with a KAN encoder leads to substantial error reductions on both benchmarks, with relative improvements of approximately 76% on Darcy Flow and 31% on CFD-1D. As shown in Fig. 3, the KAN-based model maintains significantly

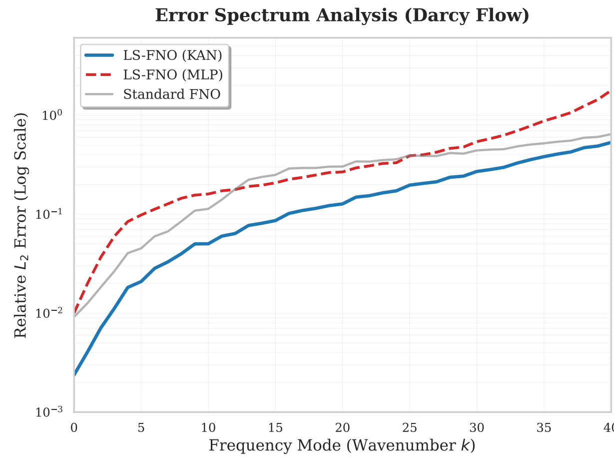

Fig. 3. **Error spectrum analysis on Darcy Flow.** LS-FNO with a KAN-based encoder exhibits consistently lower errors than both the MLP-based variant and standard FNO, with the largest gap appearing in high-frequency modes ($k > 20$), highlighting its improved spectral expressivity.

lower errors in high-frequency modes, whereas the MLP-based variant degrades rapidly as the frequency increases. This suggests that spline-based encoding provides a more suitable inductive bias for continuous spectral parameterization.

b) Latent rank analysis.: Increasing the latent rank from $r = 4$ to $r = 16$ yields clear performance gains, while further increasing r provides only marginal improvements. This behavior indicates that the spectral kernels can be well-approximated by a compact latent representation, with $r = 16$ offering a favorable trade-off between accuracy and parameter efficiency.

c) Sensitivity to architectural hyperparameters.: LS-FNO exhibits moderate sensitivity to both the KAN grid size and the model width. Larger grid sizes generally improve performance, with diminishing returns beyond $G \approx 32$. The optimal model width depends on the task: while smoother Darcy Flow saturates at $W = 32$, the shock-dominated CFD-1D benefits from wider channels, achieving its lowest error at $W = 40$.

D. Complexity and Efficiency Analysis

We compare the computational efficiency of LS-FNO with standard FNO, an enlarged variant FNO$^+$ (with increased modes and width), and the amortized baseline AM-FNO. Table IV reports the parameter count, training time per epoch, and relative L_2 error on the CFD-2D benchmark.

TABLE IV
EFFICIENCY COMPARISON ON THE CFD-2D BENCHMARK.

Model	Params (M)	Time (s/epoch)	L_2 Error
FNO	2.40	**11.4**	5.36e-3
FNO$^+$	9.58	17.5	5.04e-3
AM-FNO	2.25	24.6	2.70e-3
LS-FNO (Ours)	**2.21**	19.8	**2.23e-3**

Overall, LS-FNO achieves a favorable trade-off between accuracy and efficiency. Compared to the parameter-heavy FNO^+, LS-FNO attains substantially lower error (2.23e-3 vs. 5.04e-3) while using only **23%** of the parameters, corresponding to a **77%** reduction in model size. This suggests that increasing spectral resolution and width alone is a less effective strategy than learning a compact latent spectral representation.

In terms of computational cost, LS-FNO is slower than the lightweight FNO due to the additional spline-based computations, but it is notably more efficient than AM-FNO. Specifically, LS-FNO reduces the training time per epoch by approximately **20%** (from 24.6s to 19.8s) while achieving lower error, indicating that the proposed latent spectral generator provides a more efficient amortization of spectral parameters.

VI. Conclusion

We introduced **LS-FNO**, which preserves the FFT–multiply–IFFT operator form of FNO while replacing discrete mode-wise spectral weights with a *continuous* frequency-conditioned generator. A low-rank latent spectral manifold and a **KAN**-based encoder–decoder tie parameters across frequencies, improving scalability and high-frequency expressivity. Experiments on six benchmarks show consistent gains over strong baselines, especially on compressible CFD, with improved parameter efficiency and competitive training cost. Future work includes extending to 3D turbulence and multi-physics pretraining.

Acknowledgments

This work was supported by the National Science and Technology Major Project (2022ZD0117805), by the National Natural Science Foundation of China under Grant No. 92370113, and by the Guangdong Provincial Key Laboratory of Mathematical Foundations for Artificial Intelligence (2023B1212010001).

References

[1] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[2] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annual review of fluid mechanics*, vol. 52, no. 1, pp. 477–508, 2020.

[3] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli *et al.*, "Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators," *arXiv preprint arXiv:2202.11214*, 2022.

[4] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," *arXiv preprint arXiv:2010.08895*, 2020.

[5] M. Raissi, P. Perdikaris, and G. 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*, vol. 378, pp. 686–707, 2019.

[6] S. Wang, H. Wang, and P. Perdikaris, "Learning the solution operator of parametric partial differential equations with physics-informed deeponets," *Science advances*, vol. 7, no. 40, p. eabi8605, 2021.

[7] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via deeponet based on the universal approximation theorem of operators," *Nature machine intelligence*, vol. 3, no. 3, pp. 218–229, 2021.

[8] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Neural operator: Learning maps between function spaces with applications to pdes," *Journal of Machine Learning Research*, vol. 24, no. 89, pp. 1–97, 2023.

[9] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Neural operator: Graph kernel network for partial differential equations," *arXiv preprint arXiv:2003.03485*, 2020.

[10] J. Brandstetter, D. Worrall, and M. Welling, "Message passing neural pde solvers," *arXiv preprint arXiv:2202.03376*, 2022.

[11] T. Tripura and S. Chakraborty, "Wavelet neural operator: a neural operator for parametric partial differential equations," *arXiv preprint arXiv:2205.02191*, 2022.

[12] J. Brandstetter, R. v. d. Berg, M. Welling, and J. K. Gupta, "Clifford neural layers for pde modeling," *arXiv preprint arXiv:2209.04934*, 2022.

[13] N. Kovachki, S. Lanthaler, and S. Mishra, "On universal approximation and error bounds for fourier neural operators," *Journal of Machine Learning Research*, vol. 22, no. 290, pp. 1–76, 2021.

[14] A. Tran, A. Mathews, L. Xie, and C. S. Ong, "Factorized fourier neural operators," *arXiv preprint arXiv:2111.13802*, 2021.

[15] Z. Xiao, S. Kou, Z. Hao, B. Lin, and Z. Deng, "Amortized fourier neural operators," *Advances in Neural Information Processing Systems*, vol. 37, pp. 115 001–115 020, 2024.

[16] M. A. Rahman, Z. E. Ross, and K. Azizzadenesheli, "U-no: U-shaped neural operators," *arXiv preprint arXiv:2204.11127*, 2022.

[17] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, and S. M. Benson, "U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow," *Advances in Water Resources*, vol. 163, p. 104180, 2022.

[18] M. Kim, J. Park, and J. Shin, "Efficient fourier neural operators by group convolution and channel shuffling," in *ICLR 2024 Workshop on AI4DifferentialEquations In Science*, 2024.

[19] S. Cao, "Choose a transformer: Fourier or galerkin," *Advances in neural information processing systems*, vol. 34, pp. 24 924–24 940, 2021.

[20] Z. Li, K. Meidani, and A. B. Farimani, "Transformer for partial differential equations' operator learning," *arXiv preprint arXiv:2205.13671*, 2022.

[21] Z. Hao, Z. Wang, H. Su, C. Ying, Y. Dong, S. Liu, Z. Cheng, J. Song, and J. Zhu, "Gnot: A general neural operator transformer for operator learning," in *International conference on machine learning*. PMLR, 2023, pp. 12 556–12 569.

[22] J. Guibas, M. Mardani, Z. Li, A. Tao, A. Anandkumar, and B. Catanzaro, "Adaptive fourier neural operators: Efficient token mixers for transformers," *arXiv preprint arXiv:2111.13587*, 2021.

[23] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 5301–5310.

[24] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.

[25] Z. Li, D. Z. Huang, B. Liu, and A. Anandkumar, "Fourier neural operator with learned deformations for pdes on general geometries," *Journal of Machine Learning Research*, vol. 24, no. 388, pp. 1–26, 2023.

[26] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar, "Physics-informed neural operator for learning partial differential equations," *ACM/IMS Journal of Data Science*, vol. 1, no. 3, pp. 1–27, 2024.

[27] M. Takamoto, T. Praditia, R. Leiteritz, D. MacKinlay, F. Alesiani, D. Pflüger, and M. Niepert, "Pdebench: An extensive benchmark for scientific machine learning," *Advances in neural information processing systems*, vol. 35, pp. 1596–1611, 2022.

[28] H. Wu, T. Hu, H. Luo, J. Wang, and M. Long, "Solving high-dimensional pdes with latent spectral models," *arXiv preprint arXiv:2301.12664*, 2023.

[29] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[30] Loshchilov, Ilya and Hutter, Frank, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[31] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.