Spectral-Refiner

Accurate Fine-Tuning of Spatiotemporal Fourier Neural Operator for Turbulent Flows

Shuhao Cao[1] Francesco Brarda[2] Ruipeng Li[3] Yuanzhe Xi[2]

[1]UMKC [2]Emory [3]LLNL

The Thirteenth International Conference on Learning Representations

Spatiotemporal Operator Learning

Setting the Stage

Toy model: approximating Navier-Stokes equations in a 2D periodic box

- (Velocity) Find $\boldsymbol{u} \in \boldsymbol{H}^1(\mathbb{T}^2) \cap \{\boldsymbol{v} \in \boldsymbol{H}(\mathrm{div}) : \nabla \cdot \boldsymbol{v} = 0\}$

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \alpha \boldsymbol{u} = \boldsymbol{f},$$

- (Vorticity-Streamfunction) Find $\omega, \psi \in H^1(\mathbb{T}^2)$

$$\partial_t \omega + (\nabla^\perp \psi) \cdot \nabla \omega - \nu \Delta \omega + \alpha \omega = 0, \quad \omega + \Delta \psi = 0.$$

A long and rich history of numerical analysis for the NSE

- Projection schemes: Chorin (1968), Shen (1992).
- (Pseudo) Spectral: Orszag (1971), Orszag (1972), Tadmor (1987), Ku, Taylor, and Hirsh (1987), Shen (1994).
- Finite element: Girault and Raviart (1986), Brezzi and Fortin (1991), Temam (1995).
- Millennium Prize problem: T. Y. Hou (2009), Luo and T. Y. Hou (2014), J. Chen, T. Y. Hou, and Huang (2022).

Numerical Methods for NSE: Nonlinear Convection

- Implicit for the diffusion term, explicit for the convection term (subject to the CFL).
  - Example: Crank-Nicolson with step size τ,

$$\left(I - \frac{\tau}{2}\nu L\right)\omega^{\text{New}} = \left(I + \frac{\tau}{2}\nu L\right)\omega - \tau(\nabla^{\perp}(-\Delta)^{-1}\omega) \cdot \nabla\omega.$$

  - τ has to be small enough to be comparable $\mathcal{O}(\Delta x / \|\omega(t,\cdot)\|_{\infty})$.
- Aliasing error caused by the nonlinear term in spectral-based collocation or Galerkin methods: the modes extend "outside" of the approximation space on a fixed grid

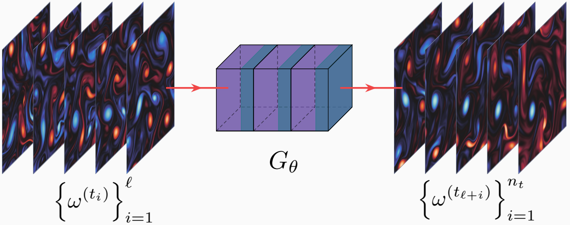
$$\mathcal{S}_n = \operatorname{span}\left\{e^{i\boldsymbol{k}\cdot\boldsymbol{x}} : \boldsymbol{k} := 2\pi(k_j),\ -n/2 \le k_j \le n/2 - 1\right\} / \mathbb{R}.$$

  - "$3/2$-rule" applied to the modes of the approximation, higher-order Fourier smoothing[1], implicit residual smoothing[2].

[1] J. Goodman, T. Hou, and E. Tadmor (1994). "On the stability of the unsmoothed Fourier method for hyperbolic equations". In: *Numerische Mathematik*.

[2] A. Lerat, J. Sidès, and V. Daru (1982). "An implicit finite-volume method for solving the Euler equations". In: *Eighth International Conference on Numerical Methods in Fluid Dynamics*.

Learning Maps between Bochner Spaces


$\left\{\omega^{(t_i)}\right\}_{i=1}^{\ell}$
G_θ
$\left\{\omega^{(t_{\ell+i})}\right\}_{i=1}^{n_t}$

Problem of interest

Construct an operator-valued neural network G_θ to “approximate” G:

$$G : L^2(t_1, t_\ell; \mathcal{V}) \to L^2(t_{\ell+1}, t_{\ell+n_t}; \mathcal{V}), \quad \{\omega(t,\cdot)\}_{t\in(t_1,t_\ell)} \mapsto \{\omega(t,\cdot)\}_{t\in(t_{\ell+1},t_{\ell+n_t})},$$

$$G_\theta : \mathcal{S}_\ell \to \mathcal{S}_{n_t}, \quad \mathbb{R}^{n\times n\times \ell} \ni \mathbf{w}_{\text{in}} \mapsto \mathbf{w}_{\text{out}} \in \mathbb{R}^{n\times n\times n_t},$$

- $\mathcal{V} := H^1(\mathbb{T}^2)$ or $\{\boldsymbol{v} \in \boldsymbol{H}^1(\mathbb{T}^2) : \nabla\cdot\boldsymbol{v} = 0\}$.
- $\mathcal{S}_n \simeq \prod_{j=1}^{n} \mathcal{S}$, where $\mathcal{S}$ is a finite-dimensional approximation space of $\mathcal{V}$.
- n, ℓ, n_t can all vary for training and evaluation.

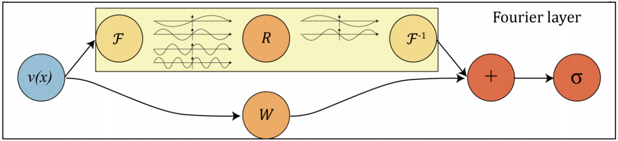
Learning Operators between Function Spaces

Neural operators (operator-valued NN) and operator learning:

$$\mathcal{L}_\theta : \mathbb{R}^{n\times n\times d_{\text{in}}} \to \mathbb{R}^{n\times n\times d_{\text{out}}}$$

- How the “basis” (“frames”) are constructed and how are they aggregated?
    - “Latent” frames/basis are (nonlinear) universal approximators, and then are linearly aggregated through coefficients independent of the current latent space, e.g., DeepONet: Lu et al. (2021), S. Wang, H. Wang, and Perdikaris (2021); Fourier Neural Operator: Z. Li, Kovachki, et al. (2021), and many others.
    - Frames/basis are linear projections of the current latent representations, then aggregated nonlinearly (input-dependent kernel integral). Nonlocal kernel: Gilboa and Osher (2007). Transformer/Attention: C. (2021), Z. Li, Meidani, and Farimani (2023), Hao et al. (2023), Bartolucci et al. (2024), and many others.
- ROM, POD. Burkardt, Gunzburger, and Lee (2006), Xiao et al. (2015).
- Neural super-resolution operator. Spatial: Kochkov et al. (2021); temporal: Sun, Yang, and Yoo (2023).
- Denoising diffusion/Gaussian processes: Y. Chen, Hosseini, Owhadi, and Stuart (2021), Lippe et al. (2023).

Fourier Neural Operator



Source: Figure 1 from the FNO paper by the Caltech applied math group.[3]

- The vector-valued kernel matrix $R_\theta(x)$ in the so-called spectral convolution operator **SpConv** is “learned” from data

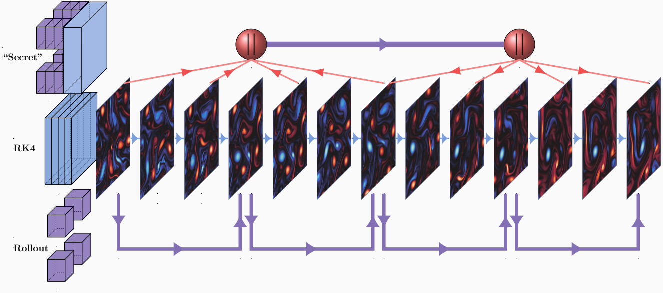
$$v^{\text{out}}(x) = \int_\Omega \kappa_\theta(x-y)v(y)dy = \mathcal{F}^{-1}\mathcal{F}\left(\int_\Omega \kappa_\theta(x-y)v(y)dy\right)$$

$$= \mathcal{F}^{-1}\left(\mathcal{F}(\kappa_\theta(x-\cdot))\cdot\mathcal{F}(v)\right)(x) \approx: \mathcal{F}^{-1}(R_\theta(x)\cdot\mathcal{F}(v))(x)$$

- A frequency truncation in $R_\theta(\cdot)$: robust learning capacity for lower modes.
- The final operator-valued NN features a “lifting” operator at the beginning and “channel reduction” at the end.

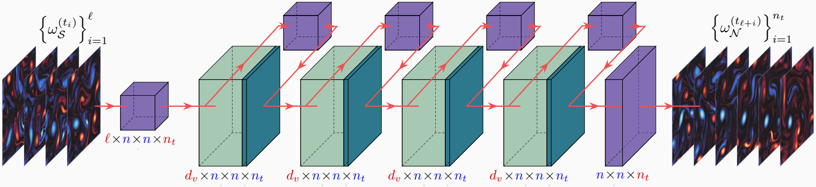
[3] Z. Li, N. B. Kovachki, et al. (2021). “Fourier Neural Operator for Parametric Partial Differential Equations”. In: *International Conference on Learning Representations*.

Toward Spatiotemporal Neural Operators



- RK4: de-aliasing filter needed, small time steps bounded by the CFL condition $\|\boldsymbol{u}(t,\cdot)\|_\infty \frac{\Delta t}{\Delta x} < c$, e.g., marching 1000 steps to move 1 unit time forward.
- Autoregressive Neural Operators ("roll-out"): using previous evaluation(s) as input, large time steps (e.g., $\delta t = 50\Delta t$). However, there are no stability guarantees and the errors are huge $\mathcal{O}(1\times 10^{-2})$.
- **Goal**: construct a spatiotemporal operator-valued NN with arbitrary spatial- or temporal-grid sizes input/output and reduce its error.

Fourier Neural Operator



$\left\{\omega_{\mathcal{S}}^{(t_i)}\right\}_{i=1}^{\ell}$
$\ell \times n \times n \times n_t$
$d_v \times n \times n \times n_t$
$d_v \times n \times n \times n_t$
$d_v \times n \times n \times n_t$
$d_v \times n \times n \times n_t$
$n \times n \times n_t$
$\left\{\omega_{\mathcal{N}}^{(t_{\ell+i})}\right\}_{i=1}^{n_t}$

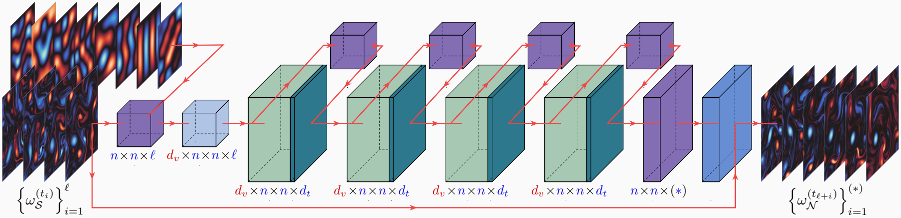
The architectural schematics of FNO for NSE: red represents fixed dimensions; blue represents dimensions that accept arbitrary-sized discretizations. 🟩: spectral convolution layer ; 🟪: pointwise `nn.Conv3d` that works as channel expansion/reduction; 🟦: pointwise nonlinearity.

- The novelty of FNO for NSE: for the input tensor of dimension $n \times n \times n_t$

  $$\text{channel dimension } d_v \leftarrow \text{time steps in the temporal dimension } n_t.$$

  However, this renders FNO unable to represent data pair in Bochner spaces. The number of parameters depends on the size of time discretization.
- The data for training and evaluation are prepared by applying a pointwise Gaussian normalizer that depends on the output steps n_t.

Spatiotemporal-Adapted Fourier Neural Operator 3D

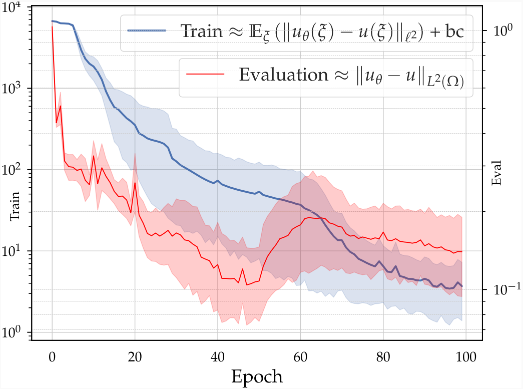

$n\times n\times \ell$
$d_v\times n\times n\times \ell$
$d_v\times n\times n\times d_t$
$d_v\times n\times n\times d_t$
$d_v\times n\times n\times d_t$
$d_v\times n\times n\times d_t$
$n\times n\times(*)$
$\left\{\omega_{\mathcal{S}}^{(t_i)}\right\}_{i=1}^{\ell}$
$\left\{\omega_{\mathcal{N}}^{(t_{\ell+i})}\right\}_{i=1}^{(*)}$

The architectural schematics of ST-FNO: 🟦: layer normalization after concatenation with positional encodings.

- Concatenation of random projection of the positional encodings with the input tensor to do channel expansion (positional encodings $\oplus$ the input data, then go through a linear layer). Thus, the channel lifting now works like a depth-wise global convolution.
- Layer normalization works as a dimension-agnostic normalizer.
- The new outprojection operator has an extra single-channel spectral convolution layer. It maps the latent time step dimension (d_t) to a given output time steps using FFT+iFFT's natural super-resolution by zero-padding the temporal steps.

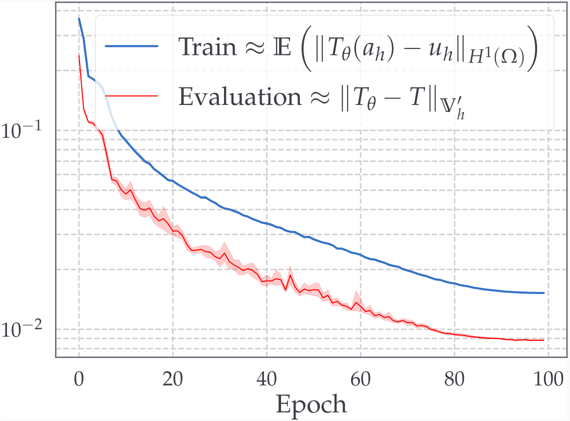
How to train Spatiotemporal FNO?

Motivations: Function Learning



An extremely difficult problem: a toy model training/evaluation relative error versus no. of epochs for a function learner to approximate $-\Delta u = 2\pi^2 \sum_{k\in\{1,4,16\}} \sin(k\pi x)\sin(k\pi y)$ in low-dimensional spaces; 1 epoch = 100 ADAM iterations, and 1 ADAM iteration randomly samples 5000 interior points and 1000 boundary points for computing the loss. Boundary penalty weight changes from 100 to 10 when the evaluation accuracy reaches 0.1. Error bars are plotted using different seeds (initialization).

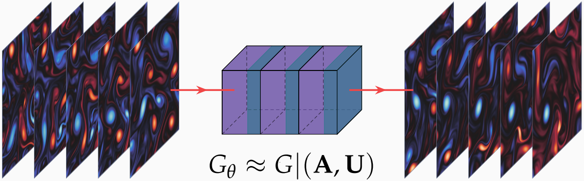
Motivations: Operator Learning



A not-so-difficult problem: “typical” convergence results for end-to-end operator-valued neural network training and evaluations for a 2D benchmark problem of porous media[4].

[4] C. (2021). “Choose a Transformer: Fourier or Galerkin”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Are Neural Operators Really Learning Operators?



$$G_\theta \approx G|(\mathbf{A}, \mathbf{U})$$

$$\mathbf{A} := \left\{\boldsymbol{w}_n \in \mathbb{R}^{n\times n\times \ell}\right\}_{m=1}^{M} \qquad \mathbf{U} := \left\{\boldsymbol{w}_n \in \mathbb{R}^{n\times n\times n_t}\right\}_{m=1}^{M}$$

> “*Given the training data pairs $\{(a_m, u_m)\}_{m=1}^{M}$, the operator learning problem is a Bayesian inverse problem with a linear or nonlinear operator as the unknown object to be inferred from data.*”

- Nelsen and Stuart (2021)[5], de Hoop, Kovachki, Nelsen, and Stuart (2023)[6].

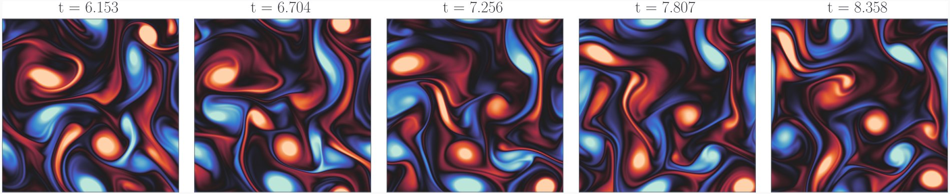
[5] N. H. Nelsen and A. M. Stuart (2021). “The random feature model for input-output maps between Banach spaces”. In: *SIAM Journal on Scientific Computing*.

[6] M. V. de Hoop, N. B. Kovachki, N. H. Nelsen, and A. M. Stuart (2023). “Convergence rates for learning linear operators from noisy data”. In: *SIAM/ASA Journal on Uncertainty Quantification*.

Statistical Property of NSE: Energy Cascade of Kolmogorov

- For 2D fluid that there is a flux of energy from smaller scales (high frequency) to larger scales (low frequency). It is called "inverse cascade"[7].
- If the dissipation of energy is caused by at larger scale vortices inducing vortex stretching via viscosity, then a stationary regime "direct cascade" is established[8].

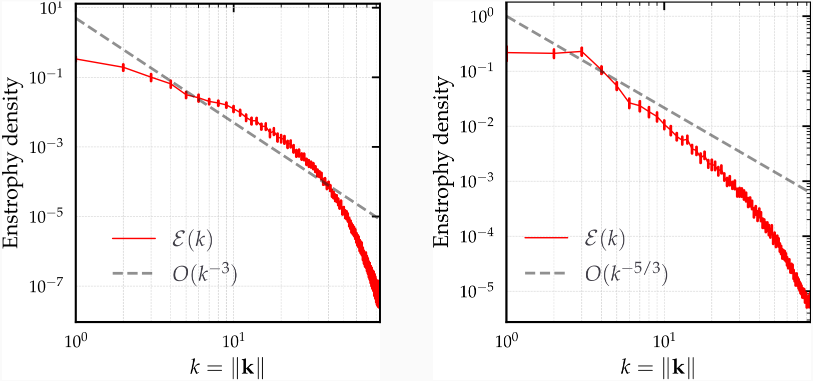
$$\mathcal{E}(t,k) = \sum_{k-\delta k\leq|\boldsymbol{k}|\leq k+\delta k} |\widehat{\nabla \times \boldsymbol{u}}(t,\boldsymbol{k})|^2 \sim \mathcal{O}(k^{-\beta}) \quad \text{after } t > t_0$$



[7] A. N. Kolmogorov (1941). "The local structure of turbulence in incompressible viscous fluid for very large Reynolds". In: *Numbers. In Dokl. Akad. Nauk SSSR*.

[8] J. C. McWilliams (1984). "The emergence of isolated coherent vortices in turbulent flow". In: *Journal of Fluid Mechanics*.

Statistical Property of NSE: Energy Cascade of Kolmogorov



The plots of $\mathcal{E}(t,k)$ for the direct cascade (left) and the inverse cascade (right). Both examples' initial conditions are sampled from fixed random fields, respectively. The error bars are plotted with $+/-$ 10 times the standard deviation from the mean to boost the visibility.

Do we really need 500 epochs of training?

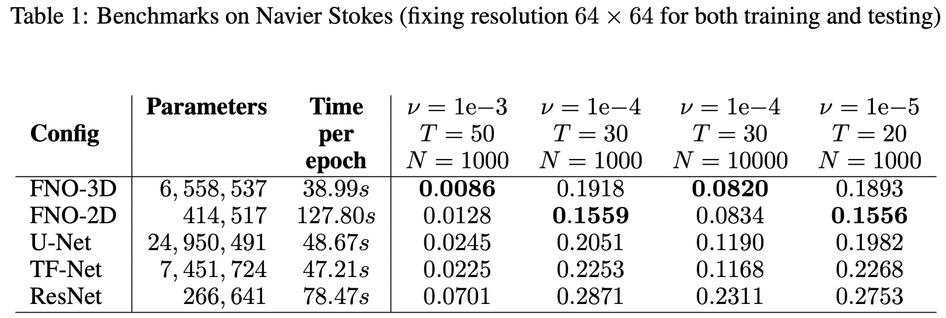


Table 1: Benchmarks on Navier Stokes (fixing resolution 64 × 64 for both training and testing)

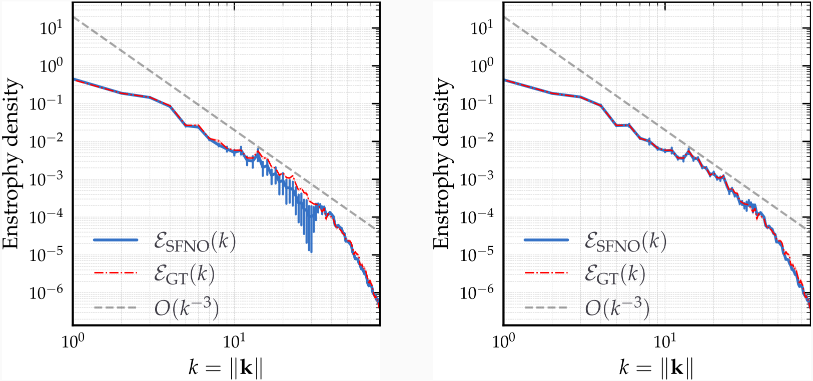
Config	Parameters	Time per epoch	$\nu = 1\text{e}{-3}$ $T = 50$ $N = 1000$	$\nu = 1\text{e}{-4}$ $T = 30$ $N = 1000$	$\nu = 1\text{e}{-4}$ $T = 30$ $N = 10000$	$\nu = 1\text{e}{-5}$ $T = 20$ $N = 1000$
FNO-3D	$6,558,537$	$38.99s$	**0.0086**	0.1918	**0.0820**	0.1893
FNO-2D	$414,517$	$127.80s$	0.0128	**0.1559**	0.0834	**0.1556**
U-Net	$24,950,491$	$48.67s$	0.0245	0.2051	0.1190	0.1982
TF-Net	$7,451,724$	$47.21s$	0.0225	0.2253	0.1168	0.2268
ResNet	$266,641$	$78.47s$	0.0701	0.2871	0.2311	0.2753

Source: Table 1 from the original FNO paper[9]. FNO3d is trained 500 epochs (500×128 mini-batch ADAM iterations). ST-FNO reaches 1×10^{-2} relative difference on a 256×256 grid with the ground truth in 10 epochs, and 6×10^{-3} after 500 epochs. More advanced coupling with Transformers[10] can drive the error down further to 4×10^{-3} level but not any further.

[9] Z. Li, N. B. Kovachki, et al. (2021). "Fourier Neural Operator for Parametric Partial Differential Equations". In: *International Conference on Learning Representations*.

[10] X. Liu, B. Xu, C., and L. Zhang (2024). "Mitigating spectral bias for the multiscale operator learning". In: *Journal of Computational Physics*.

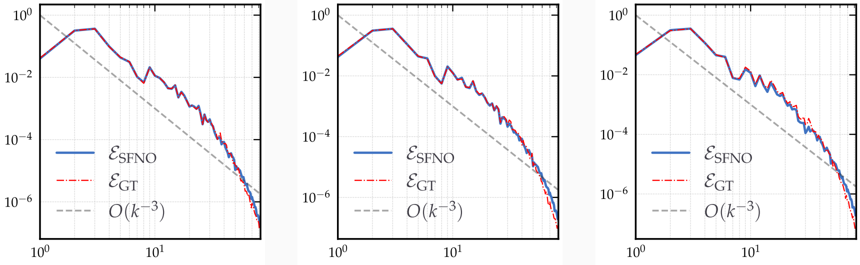
Neural Operators Learn Fast



1 epoch vs **10 epochs**: Enstrophy spectrum density comparisons to illustrate the "convergence" of ST-FNO's training. There are 10 training runs on a 64×64 grid starting from 10 different seeds. The evaluation is on a 256×256 grid for a fixed randomly chosen sample. The error bars are plotted with $+/-$ 10 times the standard deviation from the mean to boost the visibility of the convergence.

Neural Operators Learn Low-Frequency Fast

like really fast but not that accurate



Not only neural operators can learn the low-frequency modes of the data well, also known as the “spectral bias” or “frequency principle”[11], they can also learn the low modes really fast. The plots show the enstrophy spectrum density comparison for a randomly selected trajectory for a given trained ST-FNO after only 10 epochs. However, the magnitude of the error still dominates in low-frequency for modeling turbulence[12].

[11] J. Zhi-Qin Xu et al. (2020). “Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks”. In: *Communications in Computational Physics*.

[12] P. Lippe et al. (2023). “PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers”. In: *Thirty-seventh Conference on Neural Information Processing Systems*.

Motivations to Improve Accuracy

- The trainable parameters in a spectral convolution layer in FNO corresponds to the coefficients in the spectral domain.
- Use a stronger norm to train to learn low-frequency modes ($s = 0$, aggressive frequency truncation), use a weaker norm to post-process/fine-tune ($s = -1$, less truncation in the spectral domain):

$$\min_{\theta} \left\{ \| G_\theta(\boldsymbol{u}_{\mathcal{N},\mathrm{in}}) - \boldsymbol{u}_{\mathcal{N},\mathrm{out}} \|_s^2 + \|\theta\|_\alpha^2 \right\}$$

- The frequency truncation in FNO ≈ Tikhonov regularization (Hansen, Nagy, and O'Leary 2006).
- Negative Sobolev norm as an implicit regularization in inverse problems: Osher, Solé, and L. Vese (2003) and Lieu and L. A. Vese (2008), Zhu, Hu, Lou, and Yang (2024).
- The negative Sobolev norm of $f \in L^2(\mathbb{T}^2)/\mathbb{R}$ can be handily computed using FFT, and this negative norm happens to be the correct functional norm to measure the residual: $R(\boldsymbol{u}_{\mathcal{N}})(\boldsymbol{v}) := \langle R(\boldsymbol{u}_{\mathcal{N}}), \boldsymbol{v} \rangle$

$$\|f\|_{\mathcal{H}'} \simeq |f|_{-1} := \sum_{\boldsymbol{k} \in 2\pi\mathbb{Z}_n^2 \setminus \{\boldsymbol{0}\}} |\boldsymbol{k}|^{-2} \left|\hat{f}(\boldsymbol{k})\right|^2.$$

New Loss

Theorem (Reliable and efficient *a posteriori* error estimation)

Under certain smoothness assumptions for $\boldsymbol{u} \in \mathcal{H}$*, if the fine-tuning problem can be solved exactly, and the output is* $\boldsymbol{u}_{\mathcal{N}}$ *that is divergence-free, for* $m = \ell+1, \cdots, \ell+n_t-1$, $\mathcal{T}_m := [t_{m+1}, t_{n_t}]$, *the PDE residual is:*

$$R(\boldsymbol{u}_{\mathcal{N}}) := \boldsymbol{f} - \partial_t \boldsymbol{u}_{\mathcal{N}} - (\boldsymbol{u}_{\mathcal{N}} \cdot \nabla)\boldsymbol{u}_{\mathcal{N}} + \nu \Delta \boldsymbol{u}_{\mathcal{N}}, \text{ and } R(\boldsymbol{u}_{\mathcal{N}})(\boldsymbol{v}) := \langle R(\boldsymbol{u}_{\mathcal{N}}), \boldsymbol{v} \rangle .$$

The functional norm of R *is efficient:*

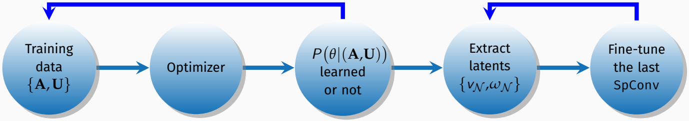
$$\|R(\boldsymbol{u}_{\mathcal{N}})\|^2_{L^2(\mathcal{T};\mathcal{V}')} \lesssim \|\boldsymbol{u} - \boldsymbol{u}_{\mathcal{N}}\|^2_{L^2(\mathcal{T};\mathcal{V})} + \|\partial_t(\boldsymbol{u} - \boldsymbol{u}_{\mathcal{N}})\|^2_{L^2(\mathcal{T};\mathcal{V}')},$$

and when $\boldsymbol{u}_{\mathcal{N}}$ *is sufficiently close,*

$$\|\boldsymbol{u} - \boldsymbol{u}_{\mathcal{N}}\|^2_{L^\infty(\mathcal{T}_m;L^2)} + \|\boldsymbol{u} - \boldsymbol{u}_{\mathcal{N}}\|^2_{L^2(\mathcal{T}_m;H^1)} \leq \|(\boldsymbol{u} - \boldsymbol{u}_{\mathcal{N}})(t_m, \cdot)\|^2_{H^1} + C \int_{\mathcal{T}_m} \|R(\boldsymbol{u}_{\mathcal{N}})(t, \cdot)\|^2_{\mathcal{V}'} \, \mathrm{d}t.$$

- Not bounded by local adaptive mesh refinement, the loss can be simply evaluated globally in the spectral domain to refine the spectral basis, similar to ROM (Patera and Rozza 2007).

New Paradigm

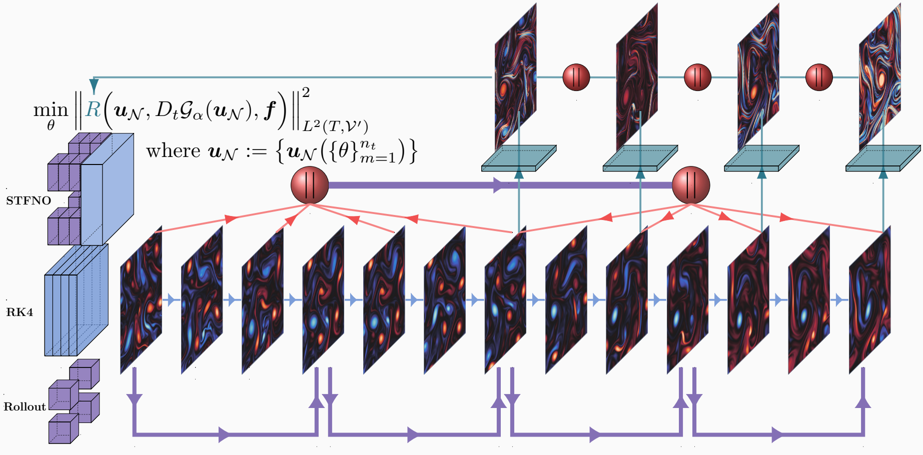

Training data $\{\mathbf{A},\mathbf{U}\}$
Optimizer
$P(\theta|(\mathbf{A},\mathbf{U}))$ learned or not
Extract latents $\{v_{\mathcal{N}},\omega_{\mathcal{N}}\}$
Fine-tune the last SpConv

- Train the neural operator only for a few epochs (e.g., 10) using a strong norm as the loss until it learns the frequency signature of the data (e.g., the energy cascade of the NSE).
- Extract the latent representation up to the last **LINEAR SpConv** layer, such that for $i = 1, \cdots, n_t$

$$\boldsymbol{u}^{(t_{\ell+i})} = \boldsymbol{u}^{(t_\ell)} + Q_\theta(\mathbf{v}_{\text{latent}}).$$

- Fine-tune this layer **ONLY** using an optimizer with a weaker norm.
- $\partial_t \boldsymbol{u}_{\mathcal{N}}$ is approximated using an auto-differentiable IMEX solver with extremely fine time steps.
- Searching for the best possible θ^* under a functional norm (negative norm) is equivalent to solving a variational problem in the positive Sobolev norm.

New Paradigm



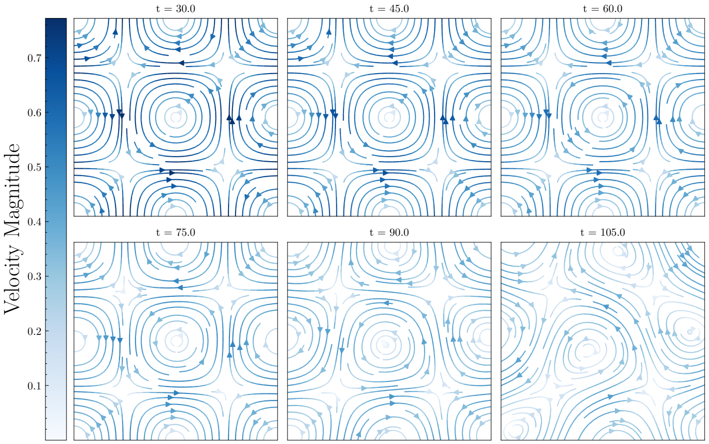
$$\min_{\theta} \left\| R\Big(\boldsymbol{u}_{\mathcal{N}}, D_t\mathcal{G}_{\alpha}(\boldsymbol{u}_{\mathcal{N}}), \boldsymbol{f}\Big) \right\|^2_{L^2(T,\mathcal{V}')}$$

$$\text{where } \boldsymbol{u}_{\mathcal{N}} := \left\{ \boldsymbol{u}_{\mathcal{N}}\big(\{\theta\}_{m=1}^{n_t}\big) \right\}$$

Schematics comparison: the new method shares striking mathematically resemblance with a parallel-in-time two-grid method. On a “coarse” temporal grid without marching 100 steps, ST-FNO gives the best possible “initial” guess for implicit residual smoothing.

Numerical Experiments

Taylor-Green Vortex



True velocity $\boldsymbol{u}(t,x,y) = e^{-2\kappa^2 \nu t} \begin{pmatrix} \sin(\kappa x)\cos(\kappa y) \\ -\cos(\kappa x)\sin(\kappa y) \end{pmatrix}$ with $\nu = 10^{-2}$.

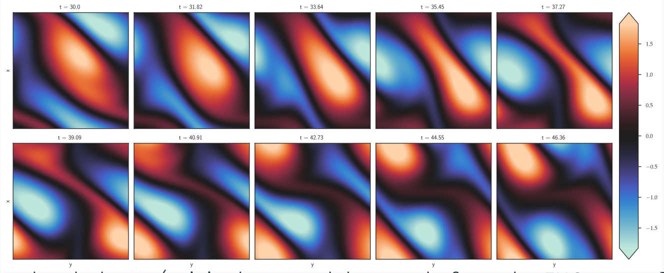
Taylor-Green Vortex

- Training dataset: 10 samples with $\kappa = 1, \ldots, 10$; evaluation: 1 sample with $\kappa = 11$.
- Trajectories are randomly sampled before the breakdown phase.
- Ground truth: pseudo-spectral discretization, second-order Runge-Kutta for the explicit part, Crank-Nicolson with $\Delta t = 10^{-4}$, on a 256×256 grid.
- Training is done on 64×64 for 5 epochs, the evaluation is on 256×256, time step is 10 for training, 40 for evaluation.

Results for Taylor-Green vortex example $\varepsilon := \omega_{\text{true}} - \omega_{\mathcal{N}}$, the errors are reported at the final time step.

	Evaluation after training		After fine-tuning	
	$\|\varepsilon\|_{L^2}$	$\|R\|_{-1,n}$	$\|\varepsilon\|_{L^2}$	$\|R\|_{-1,n}$
FNO3d	1.84×10^{-1}	5.40×10^{-1}	N/A	N/A
ST-FNO3d	1.94×10^{-1}	2.18×10^{-1}	1.24×10^{-7}	3.21×10^{-7}
PS+RK2 (GT)	5.91×10^{-6}	1.16×10^{-5}	N/A	N/A

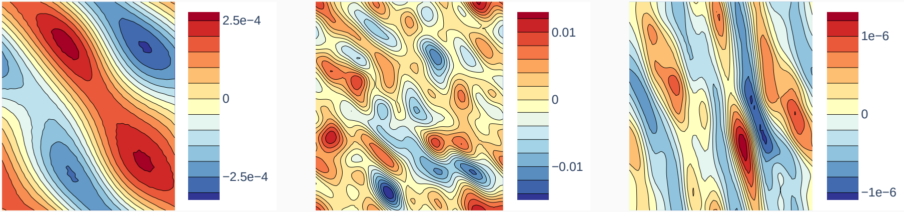
Forced Turbulence: FNO Data



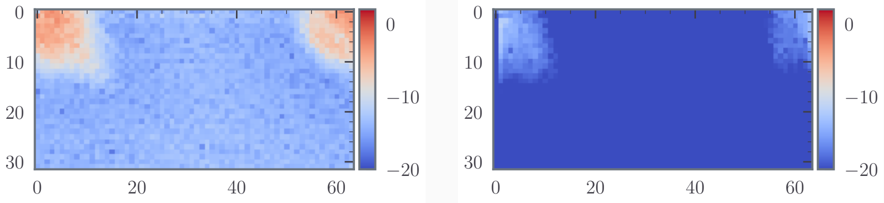
Results for forced turbulence (original toy model example from the FNO paper, $\mathrm{Re} = 1000$), small numbers of vortices. 10 epochs of training. $\varepsilon := \omega_{\mathcal{S}} - \omega_{\mathcal{N}}$ where $\omega_{\mathcal{S}}$ is a reference solution on 256×256 grid.

	Evaluation after training		After fine-tuning	
	$\|\varepsilon\|_{L^2}$	$\|R\|_{-1,n}$	$\|\varepsilon\|_{L^2}$	$\|R\|_{-1,n}$
FNO3d 100 ep	1.31×10^{-2}	1.30×10^{-2}	N/A	N/A
ST-FNO3d 10 ep + L^2 FT	1.02×10^{-2}	1.27×10^{-2}	2.82×10^{-4}	2.78×10^{-5}
ST-FNO3d 10 ep + H^{-1} FT	–	–	3.16×10^{-4}	4.59×10^{-7}

Forced Turbulence: FNO Data



Contours plots of pointwise values of residuals for the ground truth streamfunction, ST-FNO inference, and ST-FNO inference after fine-tuning (post-processing).



(Left) the log of the magnitude in the frequency domain of the pointwise residual after training. (Right) Same quantity after fine-tuning.

Acknowledgments

- The anonymous reviewers who helped us improve the presentation of the paper.
- Source codes and data to replicate the experiments are available at

  github.com/scaomath/torch-cfd 🤗 doi 10.57967/hf/2470
- References
  - C. , F. Brarda, R. Li, and Y. Xi (2025). "Spectral-Refiner: Accurate Fine-Tuning of Spatiotemporal Fourier Neural Operator for Turbulent Flows". In: *The Thirteenth International Conference on Learning Representations (ICLR)* cs.LG:2405.17211.
- The research of Brarda and Xi is supported by the National Science Foundation award DMS-2208412. The work of Li was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DEAC52-07NA27344 (LLNL-CONF-872321) and was supported by the LLNL-LDRD program under Project No. 24ERD033. The research of Cao is supported in part by the National Science Foundation award DMS-2309778.