Fourier Neural Operators for Rayleigh–Bénard Convection

Chelsea Maria John[1,2][0000-0003-3777-7393], Thibaut Lunet[2][0000-0003-1745-0780], Sebastian Götschel[2][0000-0003-0287-2120], Andreas Herten[1][0000-0002-7150-2505], Stefan Kesselheim[1][0000-0003-0940-5752], and Daniel Ruprecht[2][0000-0003-1904-2473]

[1] Jülich Supercomputing Centre, Jülich, Germany
{c.john,a.herten,s.kesselheim}@fz-juelich.de
[2] Hamburg University of Technology, Hamburg, Germany
{thibaut.lunet,sebastian.goetschel,ruprecht}@tuhh.de

Abstract. We propose an improved Fourier Neural Operator (FNO) for modeling two-dimensional Rayleigh–Bénard convection by predicting time increments instead of full solutions, achieving higher accuracy than a standard FNO baseline. The resulting model is compact (314k parameters, 1.26 MB) and fast (7 ms inference), while maintaining similar accuracy as demonstrated in previous benchmarks. We show that although FNOs generalize to finer meshes, accuracy remains limited by the resolution of the training data.

Keywords: Fourier Neural Operator, Rayleigh–Bénard convection

1 Introduction

Modeling turbulent convection is challenging and has applications from atmospheric flows [11] to industrial processes such as silicon wafer production [10]. A standard benchmark is Rayleigh–Bénard convection (RBC), where a fluid heated from below develops an overturning circulation [6] with the strength of turbulence governed by the Rayleigh number (Ra). At high Ra the flow becomes strongly turbulent, making high-resolution numerical simulations computationally expensive. Recent machine learning approaches, particularly Fourier Neural Operators (FNOs) [7], provide an alternative to mesh-based solvers by learning solution operators that, unlike PINNs [2], generalize across resolutions. In this work, we apply FNOs to 2D RBC using Dedalus [1] as ground truth, focusing on stable small-step predictions that preserve turbulent statistics and align with time-stepping solvers. We propose a lean FNO design suitable for use in iterative methods such as Parareal [4, 8].

Related Work. Hammoud et al. [3] show that PINNs reconstruct previous RBC states with relative L_2-errors on the order of 10^{-1} at $\mathrm{Ra} = 10^7$, while Almeida

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_40

et al. [12] achieve less than 10 % reconstruction error using an encoder–decoder DeepONet-based approach [9]. Wu et al. [14] introduce COAST, a causal operator framework with adaptive time stepping, reporting a VRMSE of 0.23 for $Ra = 10^7$ over a 5-step rollout. Straat et al. [13] apply FNO to 2D RBC at multiple Rayleigh numbers and obtain an average relative L_2-error of 0.021 on a 96×64 grid. In contrast, our 2D lean FNO model learns to predict increments rather than solutions and achieves significantly higher accuracy when predicting future solution states. It has fewer parameters and faster inference, which is beneficial when coupled with iterative numerical algorithms. However, it exhibits faster error accumulation under longer auto-regressive rollouts or larger time steps compared to 3D FNOs.

Contributions We propose a lean FNO architecture that learns the *increment* from the state at time t to a state at time $t + \Delta t$, similar to a time-stepping algorithm in a mesh-based simulation. We perform an ablation study which reveals that using multi-layer 1D convolutional scaling operators delivers better accuracy than linear layers. The performance of our lean FNO architecture in terms of accuracy, memory footprint and inference times is compared against the model by [13]. Finally, we analyze how FNO generalizes across spatial and temporal resolutions and show that it interpolates to new meshes, but that accuracy remains limited by the resolution of the training data.

2 Rayleigh-Bénard Convection

Rayleigh-Bénard convection occurs when a fluid within a confined channel is subjected to a temperature difference. In our setup, the lower plate is heated to a temperature θ_H while the upper plate is cooled to θ_C with a vertical distance (d) between the plates. This causes the cooler, denser fluid to sink and the warmer, lighter fluid to rise, generating convective rolls whose patterns are influenced by the aspect ratio of the channel and the physical properties of the fluid. The non-dimensional momentum conservation equations governing the dynamics of RBC are

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \theta \hat{\mathbf{z}} + \sqrt{\frac{\mathrm{Pr}}{\mathrm{Ra}}} \nabla^2 \mathbf{u}. \tag{1}$$

Here, $\mathbf{u} = (u, w)$ denotes the velocity vector with horizontal component u and vertical component w, p is the pressure, θ is buoyancy, and the Prandtl number Pr is the ratio of kinematic viscosity μ and thermal diffusivity κ. The Rayleigh number Ra, which characterizes the convective motion, is defined as

$$\mathrm{Ra} = \frac{\alpha \mathrm{g}(\theta_H - \theta_C) d^3}{\mu \kappa} \tag{2}$$

with α being the thermal expansion coefficient and g the acceleration due to gravity. The continuity equation for incompressible flow that ensures mass conservation is

$$\nabla \cdot \mathbf{u} = 0 \tag{3}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_40

Conservation of energy provides the following equation for the temperature

$$\frac{\partial \theta}{\partial t} + (\mathbf{u} \cdot \nabla)\theta = \frac{1}{\sqrt{\mathrm{Ra}\ \mathrm{Pr}}} \nabla^2 \theta. \tag{4}$$

3 Fourier Neural Operator

Fourier Neural Operators (FNOs) solve PDEs by learning mappings between function spaces using spectral representations [7]. The architecture consists of an input lifting layer $\mathcal{P}$ that maps the input function a from channel dimension d_a to a higher-dimensional representation v of dimension d_v, followed by multiple Fourier layers and a final projection $\mathcal{Q}$ that maps to the output u with channel dimension d_u. Each Fourier layer applies a Fourier transform $\mathcal{F}$, performs spectral convolution on a fixed set of low-frequency modes R, and transforms back via the inverse transform $\mathcal{F}^{-1}$. A local linear term Wv and a nonlinear activation σ are added to capture local and higher-frequency features beyond the truncated spectrum.

3.1 Generation of training data

Training and evaluation data were generated using Dedalus [1] on a 256×64 grid with a $RK443$ time stepper and $\Delta t = 10^{-3}$, with $\theta_H = 1$ and $\theta_C = 0$. Simulations start from random perturbations, run to $T_{init} = 100$ to reach a pseudo-steady turbulent state. Data is collected from $T_{init} = 100$ to $T_{sim} = 200$ for $\mathrm{Pr} = 1$ and $\mathrm{Ra} = 10^7$, with spectral analysis confirming adequate resolution. Ten simulations with different random seeds are used to record $\mathcal{U}(t) = [\mathbf{u}(t), \theta(t), p(t)]$ at each time step, forming 200 input–output pairs $(\mathcal{U}(t), \mathcal{U}(t+\Delta t))$ over $[100, 200]$. From 2000 samples, 80 % are used for training and 20 % for validation.

3.2 Learning Objective

FNOs are typically trained to predict the full solution of a PDE, but learning updates to the state can be more effective. We therefore compare two objectives: a **solution objective**, where the model directly predicts $\mathcal{U}(t + \Delta t)$ from $\mathcal{U}(t)$, and an **increment objective**, where it predicts the scaled update $\Delta\mathcal{U} = \Delta t^{-1}(\mathcal{U}(t + \Delta t) - \mathcal{U}(t))$. In the latter case, the solution is recovered via $\mathcal{U}(t+\Delta t) = \mathcal{U}(t) + \Delta t\, \mathcal{O}(\mathcal{U}(t))$, where $\mathcal{O}(\mathcal{U}(t))$ is the model increment prediction, making the FNO analogous to a one-step time integrator.

Both objectives are trained using a relative L_2 loss, applied either to the solution or the increment. For evaluation, we measure the relative error in the reconstructed solution. Note that loss and error coincide only for the solution objective. Since small Δt leads to nearly identical states, trivial predictions can yield low loss; therefore, we compare against an identity baseline that propagates $\mathcal{U}(t)$ unchanged (or predicts zero increment) to ensure meaningful learning beyond this naive predictor.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_40

4 Results

Training was conducted on a single NVIDIA A100 (40 GB) GPU of the JURECA-DC supercomputer, using a custom codebase [5] and data[3]. Our ablation study on $\mathcal{P}$ and $\mathcal{Q}$ (linear vs. 1D convolutional layers with $d_v = 16$) shows that increasing both depth and width improves performance and using cosine LR scheduler leads to better learning. The improved model is further evaluated to test the temporal and spatial generalization, and extended to compare against existing work.

4.1 Training

The baseline model is trained using both the solution and increment objectives for $\Delta t = 10^{-3}$. Table 1 compares the relative error for both objectives across the four components of the solution, that is velocity in the horizontal (u) and vertical (w) directions, buoyancy(θ), and pressure (p), averaged over 200 samples. Furthermore, the training loss is lower than the identity loss for increment objective but not for the solution objective.

The average relative error for the increment objective is two orders of magnitude smaller than for the solution objective. The noise introduced by the FNO when predicting the solution leads to an increase in error by about a factor of ten compared to when the FNO predicts the increment for buoyancy contours. Predicting increments substantially improves the quality of the generated solutions.

Table 1. Relative error computed for increment and solution objective.

Variable	Increment	Solution	Id Error
u	$5.7e{-5}$	$1.6e{-3}$	$2.0e{-4}$
w	$9.3e{-5}$	$1.7e{-3}$	$1.9e{-4}$
θ	$6.0e{-5}$	$1.0e{-3}$	$1.1e{-4}$
p	$2.4e{-5}$	$1.4e{-3}$	$7.2e{-5}$
Average	$\mathbf{5.8e{-5}}$	$1.4e{-3}$	$1.4e{-4}$

4.2 Improved model

Based on the results above, we propose the model configuration described in Table 2 as improvement over the baseline FNO. We train both for 11 500 epochs to predict increments $\Delta t = 10^{-3}$ using data generated as mentioned in subsection 3.1 which takes 11h on a single NVIDIA A100 GPU. Table 2 also shows the

[3] https://huggingface.co/datasets/chelseajohn/FNO-RBC2D_paper_artefacts

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_40

Table 2. Comparison of the improved and baseline FNO configuration

Component	**Improved Model**	**Baseline Model**
Objective	Increment	Solution
Kernel	FNO	FNO
Activation Function (σ)	GELU	GELU
Optimizer	Adam	Adam
LR Scheduler	Cosine	StepLR
Fourier Layers	2	2
Fourier Modes	12	12
Scaling Layer ($\mathcal{P}$,$\mathcal{Q}$)	4×1D Conv (width=$4d_v$)	1×Linear (width=d_v)
Input Channels (d_a)	4	4
Projection Channels (d_v)	16	16
Output Channels (d_u)	4	4
Total Parameters	314772	295552
Model Size (MB)	1.26	1.18
Inference time for batchsize=1 (ms)	7	5
Relative Error in u	2.4e-05	1.6e-03
Relative Error in w	3.2e-05	1.7e-03
Relative Error in θ	1.8e-05	1.0e-03
Relative Error in p	8.5e-06	1.4e-03
Average error	**2.1e-05**	1.4e-03

relative error in the four variables in (1) and (4) against the Dedalus reference on 256×64 grid for 200 samples from the validation dataset when predicting a single increment $\Delta t = 10^{-3}$.

4.3 Impact of model time step and autoregressive FNO application

To evaluate the impact of different time steps, the model is retrained for $\Delta t = 1$, 10^{-1}, 10^{-2}, in addition to 10^{-3} using the same architecture (Table 2) on 2000 samples for 11 500 epochs. Models are evaluated over time horizons of 1, 0.1, 0.01, and 0.001 starting from $t = 100$, using autoregressive rollout when the model time step is smaller than the target horizon. As shown in Table 3, shorter horizons yield lower validation errors due to reduced solution change, while autoregressive inference introduces mild error accumulation, leading to slightly higher errors than single-step predictions at matching time steps.

4.4 Resolution invariance

To evaluate generalization across mesh resolutions, the FNO trained on $\Delta t = 10^{-3}$ increments is tested on a 512×128 grid, twice the training resolution of

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_40

Table 3. Relative error under temporal generalization.

Data Timestep	Model Timestep	Model Steps	u	w	θ	p	Average Error
1	0.001	1000	$8.7e{-2}$	$9.8e{-2}$	$5.0e{-2}$	$4.0e{-2}$	$6.9e{-2}$
0.1	0.001	100	$8.5e{-3}$	$1.2e{-2}$	$7.4e{-3}$	$3.7e{-3}$	$8.0e{-3}$
0.01	0.001	10	$8.6e{-4}$	$1.3e{-3}$	$7.7e{-4}$	$3.6e{-4}$	$8.1e{-4}$
0.001	0.001	1	$8.6e{-5}$	$1.3e{-5}$	$7.8e{-5}$	$3.7e{-5}$	$8.2e{-5}$

Table 4. Relative error against the $(256, 64)$ Dedalus reference when evaluating the FNO trained on $(256, 64)$ across different meshes; the $(512, 128)$ case is compared to its own Dedalus reference. The interpolation column shows the error when $(256, 64)$ FNO outputs are upsampled to $(512, 128)$ via FFT.

Variable	(64, 64)	(256, 32)	(64, 32)	(256, 64)	(512, 128)	Interpolation
u	$7.9e{-4}$	$4.9e{-4}$	$5.3e{-4}$	$2.4e{-5}$	$1.0e{-4}$	$1.1e{-4}$
w	$4.7e{-4}$	$6.2e{-4}$	$4.6e{-4}$	$3.2e{-5}$	$1.4e{-4}$	$2.3e{-4}$
θ	$2.4e{-4}$	$2.6e{-4}$	$2.7e{-4}$	$1.8e{-5}$	$9.2e{-5}$	$2.0e{-4}$
p	$2.4e{-4}$	$3.9e{-4}$	$6.1e{-4}$	$8.5e{-6}$	$4.0e{-5}$	$4.0e{-5}$
Average	$4.4e{-4}$	$4.4e{-4}$	$4.7e{-4}$	$\mathbf{2.1e{-5}}$	$9.4e{-5}$	$1.5e{-4}$

256×64 used for $Ra = 10^7$. Errors are computed against reference Dedalus solutions and are found to be of the same order as those on the 256×64 grid (Table 4). Errors change very little if the reference solution on a 512×128 is restricted to a 256×64 mesh and this data used as input to the FNO. Evaluating the FNO on the trained resolution (256×64) and interpolating the output to the finer grid via FFT delivers very similar errors as evaluating the FNO directly on the finer mesh. This demonstrates that the FNO shows mesh invariance by interpolating to unseen resolutions without significant additional error. However, unlike numerical solvers, increasing inference resolution does not improve accuracy, which is instead determined by the resolution of the training data.

4.5 Comparison with Straat et al.

To evaluate the performance of our improved FNO architecture, we compare it against the model of Straat et al. [13] for $Ra = 5 \times 10^6$ with $Pr = 0.7$ on a 96×64 grid and boundary temperatures $\theta_H = 2$ and $\theta_C = 1$. We adopt their setup but use an architecture with 8 Fourier layers, 64 projection channels, and 16 Fourier modes to better capture the turbulent dynamics and larger time step. The model is trained on 44 940 samples from 15 Dedalus simulations and validated on 14 980 samples from 5 simulations, each run for 300 s with data sampled at $\Delta t = 0.1$ after a 200 s warm-up, and trained on JURECA-DC using 8 NVIDIA A100 GPUs for 500 epochs in 3.5 hours, reaching a training and validation loss of 0.03.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_40

For evaluation, predictions are generated over a 30 s window via 60 recursive steps with a model time step of 0.5 s. Averaged over 50 samples from 10 random starting points across 5 validation runs, our model achieves an average loss of 0.11 (Table 5), compared to 0.04 reported by Straat et al. Direct component comparison is not possible due to lack of data in Straat et al. While their FNO3D accumulates error more slowly, our model attains comparable initial accuracy despite its much smaller architecture, requiring only 132 MB (33 million, FP32 parameters) versus 3037 MB, and achieves faster inference times of 10 ms (batch size 10) and 30 ms (batch size 50) per prediction window, compared to 0.45 s on an NVIDIA A40 for Straat et al.

Table 5. Relative error solving the Straat et al. benchmark.

Variable	$t = 0.5$		$t = 30\,\mathrm{s}$	
	Error	**IdError**	**Error**	**IdError**
u	$4.6e{-2}$	$1.8e{-1}$	$2.0e{-1}$	$2.1e{-1}$
w	$4.9e{-2}$	$1.4e{-1}$	$1.8e{-1}$	$2.0e{-1}$
θ	$2.1e{-2}$	$3.4e{-2}$	$3.1e{-2}$	$3.6e{-2}$
p	$1.3e{-2}$	$3.0e{-2}$	$3.5e{-2}$	$3.7e{-2}$
Average	$\mathbf{3.2e{-2}}$	$9.8e{-2}$	$\mathbf{1.1e{-1}}$	$1.2e{-1}$

5 Conclusion

In this work, we systematically evaluate Fourier Neural Operators for two-dimensional Rayleigh–Bénard convection in the turbulent regime. We find that predicting increments rather than full states significantly improves accuracy and stability, reducing errors by up to two orders of magnitude for small time steps and mitigating noise, effectively turning the FNO into a data-driven one-step integrator. Architectural choices such as deeper 1D convolutional lifting/projection layers and a cosine learning rate scheduler further improve accuracy, yielding a compact and efficient model.

Our results show that FNOs are resolution-invariant in the sense that they interpolate to unseen meshes, but their accuracy is ultimately limited by the training data resolution rather than improved by finer inference grids. Compared to the much larger 3D FNO by Straat et al, our lean model achieves a trade-off between accuracy, memory, and inference speed, with substantially smaller footprint and faster runtime but stronger long-horizon error accumulation. These properties highlight the potential of FNOs as efficient surrogate models and as coarse propagators in space–time parallel methods such as Parareal.

Acknowledgments. This project received funding from the European High-Performance Computing Joint Undertaking (JU) under grant No. 101118139 (Horizon Europe), with

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_40

compute time on the GCS Supercomputer JURECA at JSC provided by the Gauss Centre for Supercomputing e.V.

References

1. Burns, K.J., Vasil, G.M., Oishi, J.S., Lecoanet, D., Brown, B.P.: Dedalus: A flexible framework for numerical simulations with spectral methods. Physical Review Research **2**(2), 023068 (Apr 2020)
2. Cai, S., Mao, Z., Wang, Z., Yin, M., Karniadakis, G.E.: Physics-informed neural networks (pinns) for fluid mechanics: A review (2021)
3. Hammoud, M.A., Alwassel, H., Ghanem, B., Knio, O., Hoteit, I.: Physics-informed deep neural network for backward-in-time prediction: Application to Rayleigh–Bénard convection. Artificial Intelligence for the Earth Systems **2**(1) (Jan 2023)
4. Ibrahim, A.Q., Götschel, S., Ruprecht, D.: Space-time parallel scaling of parareal with a physics-informed Fourier neural operator coarse propagator applied to the Black-Scholes equation. Proceedings of the Platform for Advanced Scientific Computing Conference p. 1–11 (Jun 2025)
5. John, C.M., Lunet, T., Götschel, S.: CFNO: Chebyshev fourier neural operators (software). https://github.com/Parallel-in-Time/cfno (2024), bSD-2-Clause license
6. Kadanoff, L.P.: Turbulent heat flow: Structures and scaling. Physics Today **54**(8), 34–39 (08 2001)
7. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A.: Fourier neural operator for parametric partial differential equations (2021)
8. Lions, J.L., Maday, Y., Turinici, G.: A "parareal" in time discretization of pde's. Comptes Rendus de l'Académie des Sciences. Série I. Mathématique **332** (01 2001)
9. Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G.E.: Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. Nature Machine Intelligence **3**(3), 218–229 (Mar 2021)
10. Oyama, H., Nieman, K., Tran, A., Keville, B., Wu, Y., Durand, H.: Computational fluid dynamics modeling of a wafer etch temperature control system. Digital Chemical Engineering **8**, 100102 (Sep 2023)
11. Shipley, D., Weller, H., Clark, P., McIntyre, W.: Two-fluid single-column modelling of Rayleigh-Bénard convection as a step towards multi-fluid modelling of atmospheric convection (2021)
12. de Sousa Almeida, J.L., Rocha, P.R.B., de Carvalho, A.M., Jr, A.C.N.: A coupled variational encoder-decoder - deepONet surrogate model for the Rayleigh-bénard convection problem. In: When Machine Learning meets Dynamical Systems: Theory and Applications (2023)
13. Straat, M., Markmann, T., Hammer, B.: Solving turbulent Rayleigh-Bénard convection using Fourier neural operators (Jan 2025)
14. Wu, Z., Wang, S., Zhang, S., He, S., Zhu, M., Jiao, A., Lu, L., van Dijk, D.: Tante: Time-adaptive operator learning via neural taylor expansion (2025)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_40