Solving Turbulent Rayleigh-Bénard Convection using Fourier Neural Operators

Michiel Straat[1], Thorben Markmann[1] and Barbara Hammer[1] *

1- Bielefeld University - Center For Cognitive Interaction Technology
Inspiration 1, 33619 Bielefeld - Germany

Abstract. We train Fourier Neural Operator (FNO) surrogate models for Rayleigh-Bénard Convection (RBC), a model for convection processes that occur in nature and industrial settings. We compare the prediction accuracy and model properties of FNO surrogates to two popular surrogates used in fluid dynamics: Dynamic Mode Decomposition (DMD) and the Linearly-Recurrent Autoencoder Network (LRAN). We regard Direct Numerical Simulations (DNS) of the RBC equations as the ground truth on which the models are trained and evaluated in different settings. The FNO performs favorably when compared to the DMD and LRAN and its predictions are fast and highly accurate for this task. Additionally, we show its zero-shot super-resolution ability for the convection dynamics. The FNO model has a high potential to be used in downstream tasks such as flow control in RBC.

1 Introduction

The fluid dynamics field benefits greatly from the application of AI techniques in several respects [1, 2]: Accelerating DNS, improving model accuracy, and developing Reduced Order Models (ROM), which are models that decompose the dynamics in its most prominent features, akin to PCA and non-linear autoencoders. In this work, we focus on surrogate models, which are models that replace the DNS for predicting future roll-outs of the system. They can be used in a purely data-driven manner on observation data without requiring equations or model parameters. Surrogate models are significantly faster than DNS, even when taking training data generation and model training into account [3]. Hence, they are highly suitable for parameter studies in dynamical systems and solving inverse problems. Additionally, their differentiability makes them applicable to implementing control schemes.

However, surrogate models are often dependent on the resolution of the training data. To address this, the FNO model was recently introduced [3] as a versatile network architecture that directly learns the complex-valued coefficients of convolutional Fourier-space filters. The main benefit of operator models is their function space representation [4], which makes them suitable for learning solution operators of Partial Differential Equations (PDE) that generalize to

*The authors acknowledge financial support by the project "SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059A), which is funded by the program "Netzwerke 2021" of the Ministry of Culture and Science of the State of North Rhine Westphalia, Germany.

different spatial resolutions. The FNO performed particularly well for learning solution operators of PDE, as was demonstrated on fluid flow described by Navier-Stokes in a chaotic regime [3].

In this work, we employ the FNO model for the first time to study its effectiveness as a surrogate for turbulent convection dynamics. Specifically, we address its effectiveness as a function of the amount of turbulent flow in the system. Convection is described by the RBC PDE, which models a fluid in a box that is heated from below, which causes the fluid to rise to the top of the box. Increasing the heat at the bottom makes these upward flows more turbulent. Convection is a widespread phenomenon in nature (atmosphere, Earth's mantle, oceans) and industrial settings (e.g. silicon waver production), and studying it using novel AI methods has the potential to improve applications in meteorology and the chemical industry. We aim at approximating a solution operator (or surrogate) for RBC up to highly turbulent regimes using the FNO and compare its performance to two models that we chose due to their popularity in fluid dynamics: DMD and LRAN [2, 5]. These methods aim to find a linear dynamical system in a latent space of system measurements that approximate the non-linear dynamics.

In the literature, modeling of the convective field in RBC was done in [6] using an autoencoder and a GRU in the latent space. Here, we focus on resolution-independent operator models for the entire state of RBC that include the fluid velocities and the temperature fields. In [7], the FNO and a DeepONet operator model in the latent space were compared in predicting the initial motion for one time unit in the RBC system starting from a no-motion initial condition. In contrast, we assume that convective cells have already formed and we model their dynamic patterns for long time windows and varying degrees of turbulence.

2 Methodology

2.1 Data generation using simulations

All the surrogate models used in this paper are trained fully data-driven on observations. Usually, those observations are taken from sensors in experiments [8], atmosphere, or industry. In this work, we rely on computer simulations of convection in 2D to generate the training data. We used a numerical solver for the RBC equations from the Shenfun [9] package on a 2D rectangular spatial domain[1]. The simulation yields images of the state over time and each "pixel" location has the local fluid's temperature value and velocity vector.

The *Rayleigh number* Ra is a key system parameter that determines the amount of convective turbulence in the system. For our study, we varied $Ra \in \{1e5, 1e6, 2e6, 5e6\}$, which resulted in four settings starting from moderate to

[1]Spatial dimensions: Horizontal $x \in [0, 2\pi]$, vertical $y \in [-1, 1]$, discretized to 96×64 uniform grid points. Bottom temperature boundary condition (BC) $T_H = 2$ and top BC $T_C = 1$. Zero velocity BC at top and bottom, periodic BCs at left and right. Solver timestep $dt = 0.1$. Our code repository provides further details and equations: `https://github.com/SAIL-project/RBC-FNO-Surrogate`.

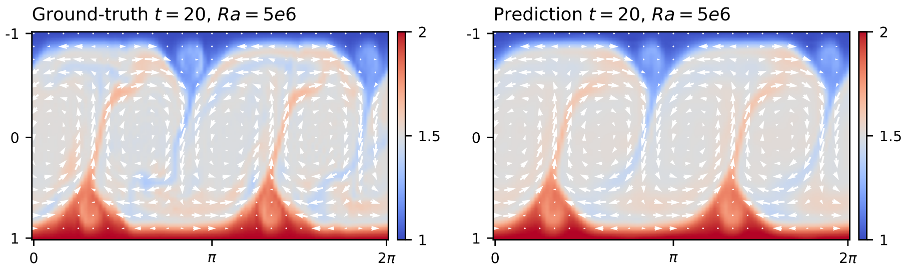

Fig. 1: *Left*: A ground truth at $t = 20$ from a random test starting point that we label $t = 0$. *Right*: the field as predicted by FNO-3D. Color: temperature field, arrows: velocity field.

high turbulence. The simulations were run for 200 initial time units to let the system transition from the no-motion state to convection and subsequently, 250 time units were recorded for the training data. In total, we generated 25 episodes for each Rayleigh number starting from slightly different random initial conditions. We divided the episodes into $[15, 5, 5]$ episodes for training, validation and testing, respectively. For the quantitative evaluation of the predictions, we calculated a Normalized Root Sum of Squared Errors (NRSSE)[2]:

$$NRSSE = ||\hat{\boldsymbol{x}} - \boldsymbol{x}|| \,/\, ||\boldsymbol{x}|| \,, \tag{1}$$

where $\hat{\boldsymbol{x}}$ is the predicted state and $\boldsymbol{x}$ is the ground truth state.

2.2 Fourier Neural Operator

Operator methods use trainable functional representations to learn a solution operator of a dynamical system in a fully data-driven way and invariant to resolution. The FNO is a specific operator architecture proposed in [3], in which the training of global convolutional filters takes place in Fourier space, where convolutions are implemented as multiplication. Similar to general neural networks, the linear processing in a layer is followed by a non-linearity, and several of these Fourier layers are used sequentially. The multiplication in Fourier space truncates the data to a predefined number of lower-frequency Fourier modes. An MLP at the start of the network lifts the state to a higher-dimensional number of hidden channels, and an MLP at the end projects back to the target dimension. See Fig. 2 in [3] for an illustrative depiction of the architecture.

We applied the FNO-3D variant that learns an operator that maps 3D functions to 3D functions by learning a series of spatiotemporal filters. We performed

[2]For the evaluation of all methods, we selected ten random starting points in each test episode for which we recurrently applied the models that performed best in the validation to predict a further time window of length 30, after which we calculated the average NRSSE with respect to the ground truth over all selected test windows.

searches over data and architecture parameters[3]. Subsequently, we extracted 3300 input-output pairs from the 15 training episodes, 1100 pairs from the five validation episodes, and 1100 pairs from the five testing episodes.

2.3 Koopman architectures

Koopman methods are popular in fluid dynamics and aim to extract a linear dynamical system in a theoretically infinite-dimensional space of *observables*, which are non-linear measurements of the system. Due to linearity, it is possible in some cases to decompose the dynamics into Koopman eigenvalues and Koopman modes (which act like eigenvectors). Next, we briefly discuss two methods that are used in fluid dynamics to extract linear dynamical systems in practice.

2.3.1 Dynamic Mode Decomposition

DMD provides a linear system on a finite-dimensional subspace, which can grow exponentially with the dimension of the state space and the complexity of the dynamics. For this reason, the kernel DMD leverages the kernel trick to implicitly compute inner products in the high-dimensional observable space [5]. The SVD is used to fit[4] the DMD on a window and the extracted Koopman modes and eigenvalues are used to predict the future evolution. Similar to ARMA models, refitting is necessary whenever making predictions.

2.3.2 Linearly Recurrent Autoencoder Network

The LRAN differs from the DMD in that it uses an autoencoder network architecture to learn 1.) the observable functions and 2.) the linear transition matrix in observable space, both simultaneously by end-to-end training using gradient descent[5]. Contrary to the FNO which takes a spatiotemporal volume as input, the LRAN takes a single input snapshot and autoregressively predicts arbitrarily-sized prediction windows. See Fig. 2 in [2] for an illustrative depiction of the architecture and [10] for an in-depth treatment of the LRAN.

3 Results and Discussion

The main result is given in Fig. 2, which shows the performance on the 30-second window prediction task averaged over randomly chosen starting points from the test episodes. We observed, also by qualitative inspection, that all methods

[3] We experimented with mapping volumes of time window $T \in \{10, 15, 20\}$ as input and output with $\Delta t = 0.5$ between individual snapshots. We performed a parameter search over the number of Fourier layers $\{8, 16, 32\}$, lower Fourier modes in the signal $\{8, 16, 32\}$, hidden channels $\{16, 32, 64\}$ and hidden units in the lifting and projection MLP $\{16, 32, 64\}$.

[4] We performed a parameter search over the width of the Gaussian kernel $\sigma \in [9.0, 100.0]$ and the length of the fitting window $T \in [20, 100]$.

[5] We performed a parameter search over the latent dimension of the autoencoder $[10, 1000]$ and the length of the windows during training $[5, 30]$. The autoencoder was pre-trained on single snapshots of the system to accelerate the training of the whole architecture later.

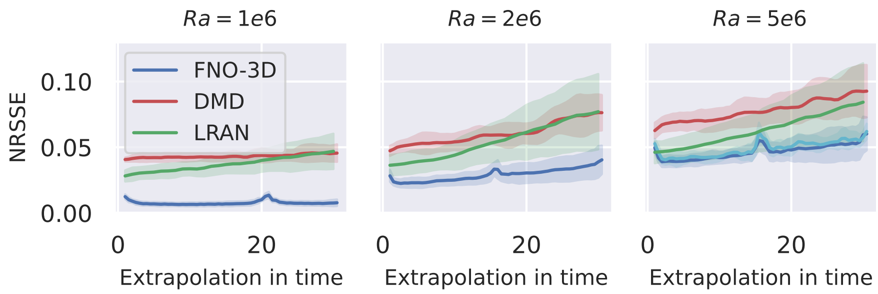

Fig. 2: Evaluation of the three models as an average error (1) computed over 50 random starting points (10 random points in each of the 5 test episodes) for increasing Rayleigh number (see figure titles). The Cyan line for $Ra = 5e6$ shows the same FNO model but evaluated on data with double the spatial resolution.

Method	Overall NRSSE	Training Time [hrs]	Prediction Time per window [s]	Memory [MB]	Resolution Invariant	Eigen Analysis
FNO	**0.021**	∼1-2	**0.45**	3037	**yes**	no
LRAN	0.042	∼1-2	0.47	**1080**	no	**yes**
DMD	0.054	-	3.16	1675	yes*	**yes**
DNS	-	-	50	23	yes*	-

Table 1: Model Comparison regarding prediction accuracy, computational complexity and their properties. *Note that DMD has to be refitted on each window, i.e., DMD does not generalize to other initial conditions or system parameters.

performed very well on the moderately turbulent cases up to $Ra = 10^6$, with the FNO-3D scoring best. For the higher turbulent cases, the error increased over the predicted window. Here, too, the FNO-3D was the most accurate, although the Koopman methods were competitive.

Table 1 lists the overall accuracy and other properties of the models. The FNO and LRAN converged in a few hours on an Nvidia A40 and generalized well to the test episodes with only the cost of forward passes, making these methods suitable for planning in model-based RL methods and parameter studies in dynamical systems. Although the FNO is 111 times faster than the DNS, its memory consumption can be substantial, as displayed by a high memory consumption compared to the other methods. A benefit of the FNO is that the trained model generalizes to other resolutions in the working phase. As we show for $Ra = 5e6$ in Fig. 2, it performs well on higher resolution data than it was trained on, without additional computational cost, while the other methods need retraining on expensive DNS simulations for higher resolutions.

The DMD is appropriate for modeling periodic patterns, which are indeed present in RBC. Although this explains the good accuracy, it should be noted that a re-fitting is necessary each time when making predictions. The fitting has a complexity of $O(n * m^2)$, where n is the number of dimensions and m is the number of snapshots in the window ($n = 3 * 64 * 96, m \leq 200$). This is likely too slow in situations where a large number of forward evaluations is necessary.

4 Conclusion

In this work, we investigated FNO-3D as a surrogate model for different levels of convective turbulence and compared the results with Koopman methods that are popular in the fluid dynamics field. All studied surrogate models were fully data-driven. At all levels of turbulence, the FNO-3D was superior concerning accuracy and zero-shot generalization to higher resolutions. The current FNO models can be used as a fast surrogate for the DNS, and can also be used in similar situations when only measurement data is available. Our future work will incorporate the fast FNO-3D predictions in a model-based RL framework for flow control in convection systems. We also aim to study more realistic 3D settings of convective flows.

These fast and accurate AI-based solutions for convection dynamics have a large potential to improve tasks in weather modeling and the chemical industry.

References

[1] Ricardo Vinuesa and Steven L. Brunton. Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, 2(6):358–366, June 2022.

[2] Thorben Markmann, Michiel Straat, and Barbara Hammer. Koopman-Based Surrogate Modelling of Turbulent Rayleigh-Bénard Convection. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2024.

[3] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

[4] M. Straat, M. Kaden, M. Gay, T. Villmann, A. Lampe, U. Seiffert, M. Biehl, and F. Melchert. Learning vector quantization and relevances in complex coefficient space. *Neural Computing and Applications*, 32(24):18085–18099, December 2020.

[5] Matthew O. Williams, Clarence W. Rowley, and Ioannis G. Kevrekidis. A kernel-based method for data-driven koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, May 2016. Publisher: Journal of Computational Dynamics.

[6] Sandeep Pandey, Philipp Teutsch, Patrick Mäder, and Jörg Schumacher. Direct data-driven forecast of local turbulent heat flux in Rayleigh-Bénard convection. *Physics of Fluids*, 34(4):045106, April 2022.

[7] Katiana Kontolati, Somdatta Goswami, George Em Karniadakis, and Michael D. Shields. Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems. *Nature Communications*, 15(1):5101, June 2024.

[8] Laurens E. Howle. Active control of Rayleigh-Bénard convection. *Physics of Fluids*, 9(7):1861–1863, July 1997.

[9] Mikael Mortensen. Shenfun - automating the spectral galerkin method. In *MekIT'17 - Ninth national conference on Computational Mechanics*, pages 273–298. International Center for Numerical Methods in Engineering (CIMNE), 2017.

[10] Samuel E. Otto and Clarence W. Rowley. Linearly Recurrent Autoencoder Networks for Learning Dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, January 2019.