FOURIER NEURAL OPERATORS FOR ARBITRARY RESOLUTION CLIMATE DATA DOWNSCALING

Qidong Yang
New York University
New York, USA
qy707@nyu.edu

Alex Hernandez-Garcia
Mila Quebec AI Institute
Montreal, Canada

Paula Harder
Fraunhofer ITWM
Kaiserslautern, Germany

Venkatesh Ramesh
Mila Quebec AI Institute
Montreal, Canada

Prasanna Sattegeri
IBM Research
New York, USA

Daniela Szwarcman
IBM Research
Brazil

Campbell D. Watson
IBM Research
New York, USA

David Rolnick
Mila Quebec AI Institute
Montreal, Canada

ABSTRACT

Running climate simulations informs us of future climate change. However, it is computationally expensive to resolve complex climate processes numerically. As one way to speed up climate simulations, neural networks have been used to downscale climate variables from fast-running low-resolution simulations. So far, all neural network downscaling models can only downscale input samples with a pre-defined upsampling factor. In this work, we propose a Fourier neural operator downscaling model. It trains with data of a small upsampling factor and then can zero-shot downscale its input to arbitrary unseen high-resolutions. Evaluated on Navier-Stokes equation solution data and ERA5 water content data, our downscaling model demonstrates better performance than widely used convolutional and adversarial generative super-resolution models in both learned and zero-shot downscaling. Our model's performance is further boosted when a constraint layer is applied. In the end, we show that by combining our downscaling model with a low-resolution numerical PDE solver, the downscaled solution outperforms the solution of the state-of-the-art high-resolution data-driven solver. Our model can be used to cheaply and accurately generate arbitrarily high-resolution climate simulation data with fast-running low-resolution simulation as input.

1 INTRODUCTION

Climate simulations are running hundreds of years ahead to help us understand how climate changes in the future. Complex physical processes inside climate dynamic systems are captured by partial differential equations (PDEs), which are extremely expensive to solve numerically. As a result, running a long-term high-resolution climate simulation is still not feasible within the foreseeable future (Balaji, 2021), even with the current fast-increasing computational power. Given the fast forward inference speed of neural networks, deep learning was applied to speed up climate simulations.

Climate simulations at low resolution are much cheaper to run than at high resolution. Therefore, there are attempts to use network networks to generate high-resolution climate variables out of their low-resolution counterpart. Such a process is named downscaling in climate science community (or super-resolution in machine learning community). Höhlein et al. (2020) used convolutional neural networks (CNNs) to downscale short-range forecasts of near-surface wind fields. A conditional generative adversarial network (GAN) was trained by Price & Rasp (2022) via a custom training procedure and augmented loss function to downscale precipitation forecasts. Groenke et al. (2020)

proposed the first unsupervised statistical downscaling method based on normalizing flows to increase the resolution of temperature and precipitation data.

Limited by classic neural networks, which map between finite-dimensional spaces, all neural network downscaling models so far have fixed input and output sizes. For a single trained model, it can only downscale input samples with a pre-defined upsampling factor. Inspired by the recent success of Fourier neural operator (Li et al., 2021, FNO) of solving PDEs regardless of resolution, we design a novel FNO zero-shot downscaling model which is able to downscale input samples to arbitrary unseen high resolution with training only once on data of a small upsampling factor. Experiments on Navier-Stokes solution data and ERA5 reanalysis (Hersbach et al., 2020) water content data show that our model achieves great performance not only on the learned downscaling (i.e., the model is trained on) tasks but also on zero-shot downscaling (i.e., the model is not trained on) tasks. The performance is even further improved when a softmax constraint layer (Harder et al., 2022) is stacked at the end of our model architecture to enforce conservation laws. Using our model to downscale low-resolution solution from a numerical Navier-Stokes equation solver, the downscaled solution obtains significantly higher accuracy than that from an FNO equation solver—one of the state-of-the-art data-driven solvers. These results validate our model's potential to cheaply and accurately generate arbitrarily high-resolution climate simulation with fast-running low-resolution simulation as input.

2 Methodology

2.1 Problem Setup

Consider low-resolution input $\mathbf{a} \in \mathbb{R}^{d_a}$ and high-resolution output $\mathbf{b} \in \mathbb{R}^{d_b}$ with $d_a < d_b$. So far, neural network downscaling models are looking for a mapping $f : \mathbb{R}^{d_a} \to \mathbb{R}^{d_b}$ from low-resolution input $\mathbf{a}$ to high-resolution output $\mathbf{b}$. This formulation induces a limitation where the downscaled output resolution is fixed to be d_b. We propose the following formulation to relax this limitation to achieve arbitrary resolution downscaling.

Instead of looking for a mapping between two finite-dimensional spaces, our methodology learns a mapping from a finite-dimensional space to an infinite-dimensional space. Namely, this mapping takes in low-resolution input $\mathbf{a} \in \mathbb{R}^{d_a}$ and outputs a function $\mathbf{u} \in \mathcal{U}$ of which high-resolution observation $\mathbf{b}$ is a discretization. We denote this mapping as: $G^\dagger : \mathbb{R}^{d_a} \to \mathcal{U}$, where $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^{d_u})$ is a Banach space of functions taking values in $\mathbb{R}^{d_u}$ at each point from a bounded open set $D \subset \mathbb{R}^d$. As a result, arbitrarily high-resolution outputs can be obtained by evaluating $\mathbf{u}$ at arbitrarily many points from D.

Suppose we have observations $\{\mathbf{a}_j, \mathbf{u}_j\}_{j=1}^N$, where $\mathbf{a}_j$ is an i.i.d. low-resolution sample and $\mathbf{u}_j = G^\dagger(\mathbf{a}_j)$ is possibly corrupted with some random noise. We aim to construct a parametric map as follows to approximate $G^\dagger$:

$$G : \mathbb{R}^{d_a} \times \Theta \to \mathcal{U} \qquad \text{or equivalently,} \qquad G_\theta : \mathbb{R}^{d_a} \to \mathcal{U},\ \theta \in \Theta, \tag{1}$$

where Θ is a finite-dimensional parameter space. We hope to find a $\theta^\dagger \in \Theta$ such that $G(\mathbf{a}, \theta^\dagger) = G_{\theta^\dagger}(\mathbf{a})$ is close to $G^\dagger(\mathbf{a})$. It can be formulated as an optimization problem:

$$\theta^\dagger = \arg\min_{\theta \in \Theta} \mathbb{E}_{\mathbf{a}}[C(G(\mathbf{a}, \theta), G^\dagger(\mathbf{a}))], \tag{2}$$

where $C : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is a cost functional measuring the distance in $\mathcal{U}$.

2.2 Implementation

Here we construct G_θ as the following:

$$G_\theta(\mathbf{a}) := \mathcal{F}_\theta(I(f_\theta(\mathbf{a}))). \tag{3}$$

$f_\theta : \mathbb{R}^{d_a} \to \mathbb{R}^d$ is a vector-valued function parameterized by a neural network. $I : \mathbb{R}^d \to \mathcal{E}(D; \mathbb{R}^{d_e})$ is an interpolation operator, which interpolates the output of f_θ as a function $\mathbf{e} \in \mathcal{E}$ over domain D. $\mathcal{F}_\theta : \mathcal{E} \to \mathcal{U}$ is a functional operator parameterized by a neural operator (Li et al., 2020). In particular, I can be a very simple interpolation scheme (e.g. linear interpolation) without hurting the expressiveness of the overall model G_θ. There are two reasons for it. First, f_θ is able to learn an embedding with a high channel dimension such that the simple interpolation of it retains rather high expressiveness for the target with a low channel dimension. Second, $\mathcal{F}_\theta$ can learn highly non-linear operators to apply complicated transformations to interpolated function $\mathbf{e} = I(f_\theta(\mathbf{a}))$ despite of the simple components of $\mathcal{F}_\theta$ (Li et al., 2021).

In this work, f_θ is represented by a residual convolutional network inspired by the generator architecture of a widely used super-resolution GAN (Wang et al., 2018); an FNO is implemented for $\mathcal{F}_\theta$; and cubic interpolation is used as I. Figure 1(a) shows an illustration of the overall structure of our proposed Fourier neural operator downscaling (DFNO) model denoted by G_θ. The detailed architecture of neural network f_θ is pictured in Figure 1(b). As for the FNO architecture for $\mathcal{F}_\theta$, it is specified in the paper by Li et al. (2021).

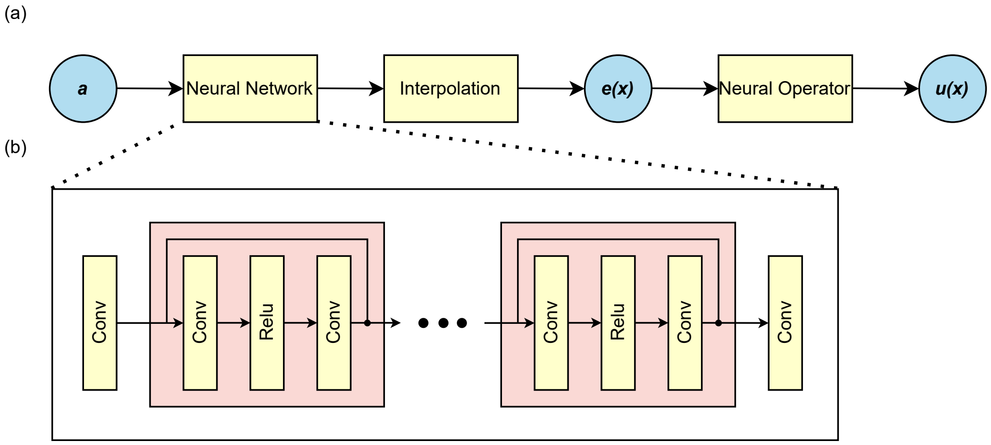


Figure 1: The upper panel shows the overall structure of our Fourier neural operator downscaling model denoted by G_θ. The low-resolution input $\mathbf{a}$ goes through a neural network f_θ and an interpolation operator I. Then an embedding function $\mathbf{e}(x)$ over domain D is returned. Finally, a neural operator $\mathcal{F}_\theta$ takes in $\mathbf{e}(x)$ and outputs the target function $\mathbf{u}(x)$ which interpolates the high-resolution observation of $\mathbf{a}$. The lower panel shows the detailed architecture of f_θ. It starts and ends with a convolutional layer. In the middle, it is composed of a series of convolutional residual blocks.

3 Experiments

3.1 Downscale PDE Data

2D Navier-Stokes equation for a viscous and incompressible fluid in vorticity form is solved to construct our PDE dataset. The equation was numerically solved 10000 times with randomly sampled initial conditions at resolution 64×64. Each solution was integrated for 50 time steps with a viscosity of 10^{-4}. Out of 10000 solutions, 7000, 2000, and 1000 solutions were sampled as training, validation, and test sets. Solutions at each time step were then downsampled via average pooling to resolution 32×32 and 16×16. It forms our PDE downscaling dataset. Following implementation details specified in Section 2.2, a DFNO was constructed. It was trained on the PDE downscaling dataset of upsampling factor 2 (i.e., $16 \times 16 \rightarrow 32 \times 32$) and then evaluated at 2 times and 4 times downscaling. Two CNN and two GAN downscaling models with pre-defined upsampling factors 2 and 4 were developed to form baselines. They were trained on datasets of their corresponding upsampling factors, and outputs were then resampled via cubic interpolation to achieve desired resolution for evaluation. The downscaling performance of all models is summarized in Table 1. To our surprise, DFNO performs worse on reconstructing inputs than downscaling inputs. However, it is not a major issue because downscaling models are intended to increase input resolution. For 2 times downscaling on which DFNO was trained, DFNO shows dominant performance over baseline models in all evaluation metrics. This performance advantage persists when it comes to zero-shot (4 times) downscaling, winning models directly trained on 4 times downscaling dataset. Table 2 collects model downscaling performance with softmax constraint layer (Harder et al., 2022) applied to enforce conservation laws. The constraint layer further improves DFNO's skill and eliminates its reconstruction error. One PDE solution downscaling example by our constrained DFNO model is presented in Figure 2.

3.2 Downscale ERA5 Data

The ERA5 dataset (Hersbach et al., 2020) is a reanalysis product from the European Center for Medium-Range Weather Forecast (ECMWF) that combines model data with worldwide observations. For this work, one quantity we focus on is the total column water that describes the vertical integral of the total amount of atmospheric water content, including water vapor, cloud water, and cloud ice but not precipitation. We extract a random 128×128 patch from the global water content field of size 721×1440 at each time step. There are roughly 60,000 time steps available in total. Then 40,000 patches are randomly sampled for training and 10,000 for each validation and testing. The low-resolution counterparts are created by taking average pooling on high-resolution samples following the standard practice as in Serifi et al. (2021); Leinonen et al. (2021). It results in low-resolution samples of sizes 32×32 and 64×64. Like the previous section, a DFNO model is trained with 2 times downscaling data and tested at 1 time, 2 times, and 4 times downscaling. Its performance is also compared against two CNN and two GAN downscaling models of upsampling factors 2 and 4. The downscaling performance of all models is collected in Table 3 (without constraint layer) and Table 4 (with constraint layer). Still, DFNO shows better performance on downscaling inputs than on reconstructing inputs. In learned downscaling, DFNO has the highest skill among all baseline models. For zero-shot downscaling, MAE score of DFNO is slightly worse than that of baseline CNN-4 and GAN-2. When the constraint layer is applied, DFNO shows dominant performance in both learned and zero-shot downscaling. Figure 3 illustrates a case study on DFNO downscaling ERA5 water content data.

3.3 Downscale for PDE Integration

This section compares two ways of integrating PDE at high resolution. The first way is to solve PDE numerically at low resolution, then downscale the solution to a higher resolution. The second way is using data-driven models to predict solutions at high resolution auto-regressively. Here we use the Navier-Stokes equation from Section 3.1 as an example. Two constrained DFNO models are implemented to downscale PDE solutions at resolution 16×16. These two DFNO models are trained with 2 times and 4 times PDE downscaling data and denoted as DFNO-2 and DFNO-4, respectively. On the other hand, as for the data-driven solver, two FNO models are developed, which predict a solution one time step forward based on the solution at the previous ten time steps. They are trained with solution data at resolution 32×32 and 64×64, denoted as FFNO-32 and FFNO-64. Both of them are then evaluated at resolution 32×32 and 64×64. The solutions generated by DFNO and FFNO models are compared against ground truth numerical solutions, and the performance is summarized in Table 5. Overall, DFNO models show a significant performance advantage over FFNO models. Comparing between DFNO models, zero-shot downscaling is still not as good as learned downscaling. Solution examples generated by FFNO-64 and DFNO-2 at resolution 64×64 for five consecutive time steps are presented in Figures 4 and 5.

4 Conclusion

In this study, we design the first arbitrary resolution downscaling model for climate data using Fourier neural operator. This model is evaluated on a Navier-Stokes equation solution dataset and an ERA5 reanalysis water content dataset. We show that our model improves performance on both datasets relative to widely used CNN and GAN super-resolution architectures. It also generalizes the learned downscaling pattern to a higher upsampling factor task achieving great zero-shot downscaling performance even dominating CNN and GAN models directly trained on this task. In addition, our model's performance is further boosted when a softmax constraint layer is applied to enforce conservation laws. In the end, we compare two ways to integrate PDE at high-resolution. Combining our downscaling model with a low-resolution numerical solver, the downscaled solution has superior accuracy to that of the state-of-the-art high-resolution data-driven solver. The good performance demonstrated by our model on climate data may result from the fact that climate data are all very smooth without high-frequency Fourier components. It means those data have a succinct representation in Fourier basis and thus can be easily captured by Fourier neural operator with a truncated Fourier series. It would be interesting to explore how to modify our model to adapt to data without a succinct Fourier representation.

REFERENCES

V. Balaji. Climbing down charney's ladder: machine learning and the post-dennard era of computational climate science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194):20200085, 2021. doi: 10.1098/rsta.2020.0085. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2020.0085.

Brian Groenke, Luke Madaus, and Claire Monteleoni. ClimAlign: Unsupervised statistical downscaling of climate variables via normalizing flows. In *Proceedings of the 10th International Conference on Climate Informatics*. ACM, sep 2020. doi: 10.1145/3429309.3429318. URL https://doi.org/10.1145%2F3429309.3429318.

Paula Harder, Qidong Yang, Venkatesh Ramesh, Prasanna Sattigeri, Alex Hernandez-Garcia, Campbell Watson, Daniela Szwarcman, and David Rolnick. Generating physically-consistent high-resolution climate data with hard-constrained neural networks, 2022. URL https://arxiv.org/abs/2208.05424.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

Kevin Höhlein, Michael Kern, Timothy Hewson, and Rüdiger Westermann. A comparative study of convolutional neural network models for wind field downscaling. *Meteorological Applications*, 27 (6), nov 2020. doi: 10.1002/met.1961. URL https://doi.org/10.1002%2Fmet.1961.

Jussi Leinonen, Daniele Nerini, and Alexis Berne. Stochastic super-resolution for downscaling time-evolving atmospheric fields with a generative adversarial network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(9):7211–7223, sep 2021. doi: 10.1109/tgrs.2020.3032790. URL https://doi.org/10.1109%2Ftgrs.2020.3032790.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. arXiv, 2020. doi: 10.48550/ARXIV.2003.03485. URL https://arxiv.org/abs/2003.03485.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Ilan Price and Stephan Rasp. Increasing the accuracy and resolution of precipitation forecasts using deep generative models, 2022. URL https://arxiv.org/abs/2203.12297.

Agon Serifi, Tobias Günther, and Nikolina Ban. Spatio-temporal downscaling of climate data using convolutional and error-predicting neural networks. *Frontiers in Climate*, 3, apr 2021. doi: 10.3389/fclim.2021.656479. URL https://doi.org/10.3389%2Ffclim.2021.656479.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks, 2018. URL https://arxiv.org/abs/1809.00219.

APPENDIX

Table 1: Downscaling performance on the PDE dataset in terms of mean squared error (MSE), mean absolute error (MAE), peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM). The best scores are highlighted in bold red, second best in bold blue. The DFNO model was trained on 2 times downscaling data, then tested on 1 time, 2 times, and 4 times downscaling. CNN-2 (GAN-2) and CNN-4 (GAN-2) represent convolutional (generative adversarial) downscaling models with predefined upsampling factors 2 and 4. They were trained on datasets of their corresponding upsampling factors, whose downscaling results are then downsampled or upsampled via cubic interpolation to get desired resolution for evaluation.

Metric	Factor	DFNO	CNN-2	CNN-4	GAN-2	GAN-4	Cubic
MSE	1×	0.0146	0.0057	0.0123	**0.0056**	0.0131	**0.0000**
MSE	2×	**0.0015**	**0.0043**	0.0052	0.0045	0.0062	0.0252
MSE	4×	**0.0037**	0.0093	**0.0070**	0.0095	0.0080	0.0350
MAE	1×	0.0826	0.0524	0.0697	**0.0520**	0.0746	**0.0000**
MAE	2×	**0.0238**	**0.0397**	0.0458	0.0424	0.0534	0.1027
MAE	4×	**0.0359**	0.0579	**0.0495**	0.0601	0.0573	0.1150
PSNR	1×	40.2750	44.3504	41.0302	**44.4541**	40.7810	**154.0983**
PSNR	2×	**50.2061**	**45.7778**	44.8762	45.5806	44.2337	38.0326
PSNR	4×	**46.3361**	42.4054	**43.6083**	42.3192	43.1123	36.6248
SSIM	1×	0.9934	0.9968	0.9935	0.9963	**0.9890**	**1.0000**
SSIM	2×	**0.9981**	**0.9962**	0.9952	0.9956	0.9917	0.9741
SSIM	4×	**0.9920**	0.9842	**0.9879**	0.9835	0.9846	0.9335

Table 2: Similar to Table 1 but softmax constraint layer is applied to the output of each model.

Metric	Factor	DFNO	CNN-2	CNN-4	GAN-2	GAN-4	Cubic
MSE	1×	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
MSE	2×	**0.0011**	**0.0038**	0.0063	**0.0038**	0.0084	0.0365
MSE	4×	**0.0029**	0.0217	**0.0063**	0.0228	0.0064	0.0517
MAE	1×	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
MAE	2×	**0.0196**	**0.0363**	0.0528	0.0365	0.0627	0.1241
MAE	4×	**0.0313**	0.1032	**0.0457**	0.1058	0.0461	0.1431
PSNR	1×	151.8861	**153.3908**	152.4238	**153.3476**	152.1304	152.4239
PSNR	2×	**51.8071**	**46.2719**	44.2463	46.2266	43.0041	36.4336
PSNR	4×	**47.4375**	38.7146	**44.1036**	38.5096	44.0425	34.9377
SSIM	1×	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
SSIM	2×	**0.9987**	**0.9969**	0.9942	**0.9969**	0.9920	0.9659
SSIM	4×	**0.9937**	0.9605	**0.9894**	0.9583	0.9892	0.9108

Table 3: Downscaling performance on the ERA5 water content dataset in terms of mean squared error (MSE), mean absolute error (MAE), peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM). The best scores are highlighted in bold red, second best in bold blue. The DFNO model was trained on 2 times downscaling data, then tested on 1 time, 2 times, and 4 times downscaling. CNN-2 (GAN-2) and CNN-4 (GAN-2) represent convolutional (generative adversarial) downscaling models with predefined upsampling factors 2 and 4. They were trained on datasets of their corresponding upsampling factors, whose downscaling results are then downsampled or upsampled via cubic interpolation to get desired resolution for evaluation.

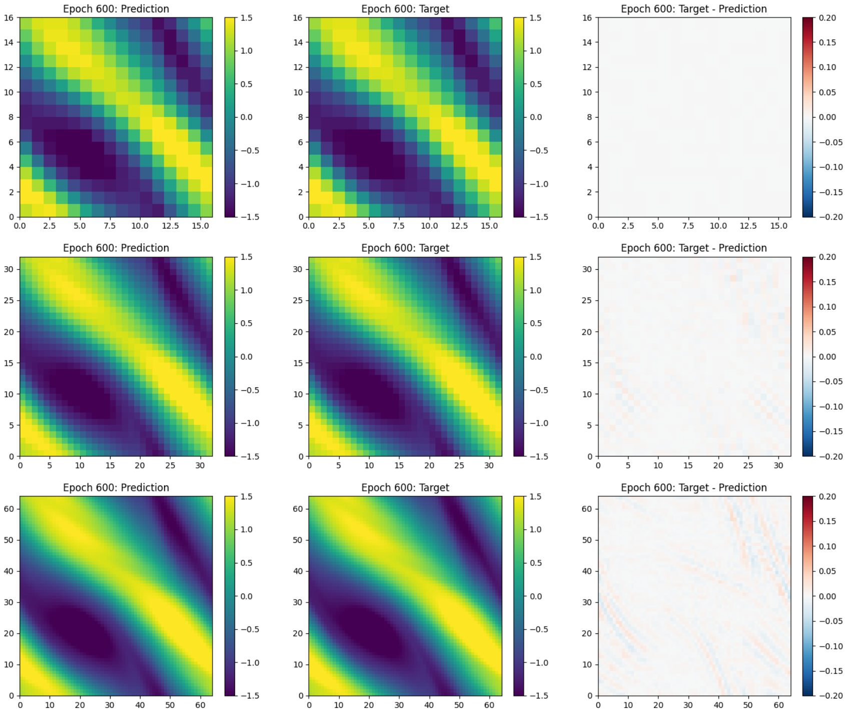
Metric	Factor	DFNO	CNN-2	CNN-4	GAN-2	GAN-4	Cubic
MSE	1×	0.2140	0.0940	0.1566	**0.0930**	0.1752	**0.0000**
MSE	2×	**0.2063**	0.2489	0.2677	**0.2474**	0.2815	0.4201
MSE	4×	**0.3628**	0.3870	**0.3851**	0.3853	0.3971	0.5954
MAE	1×	0.2896	0.1737	0.2149	**0.1731**	0.2439	**0.0000**
MAE	2×	**0.2392**	**0.2541**	0.2668	0.2542	0.2920	0.3380
MAE	4×	0.3067	0.3023	**0.3009**	**0.3022**	0.3251	0.3838
PSNR	1×	46.9630	50.5294	48.3152	**50.5795**	47.8863	**173.5160**
PSNR	2×	**48.1002**	47.2861	46.9688	**47.3111**	46.7714	45.0115
PSNR	4×	**46.0154**	45.7349	**45.7561**	45.7535	45.6330	43.8633
SSIM	1×	0.9964	**0.9982**	0.9971	**0.9982**	0.9971	**1.0000**
SSIM	2×	**0.9941**	0.9933	0.9933	**0.9934**	0.9932	0.9891
SSIM	4×	**0.9895**	0.9882	0.9887	0.9884	**0.9888**	0.9835

Table 4: Similar to Table 3 but softmax constraint layer is applied to the output of each model.

Metric	Factor	DFNO	CNN-2	CNN-4	GAN-2	GAN-4	Cubic
MSE	1×	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
MSE	2×	**0.1696**	**0.2181**	0.2896	**0.2181**	0.2964	0.8314
MSE	4×	**0.2779**	0.6054	**0.3334**	0.6118	0.3355	1.1552
MAE	1×	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
MAE	2×	**0.2250**	0.2427	0.3055	**0.2422**	0.3116	0.5318
MAE	4×	**0.2768**	0.4383	**0.2838**	0.4386	0.2851	0.5950
PSNR	1×	164.1793	**170.2039**	166.2301	**169.6977**	165.4083	161.0459
PSNR	2×	**48.9508**	**47.8585**	46.6269	**47.8585**	46.5268	42.0471
PSNR	4×	**47.1723**	43.7915	**46.3821**	43.7464	46.3550	40.9850
SSIM	1×	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
SSIM	2×	**0.9952**	0.9937	0.9919	**0.9938**	0.9917	0.9778
SSIM	4×	**0.9910**	0.9792	**0.9893**	0.9793	0.9892	0.9639

Table 5: This table compares two ways of solving the Navier-Stokes equation at high resolution concerning mean squared error (MSE) and mean absolute error (MAE). First way: solve the equation numerically at low resolution (16 × 16); then downscale the solution to 32 × 32 and 64 × 64 by constrained DFNO models. Second way: use data-driven FFNO models to auto-regressively predict solutions at resolutions 32 × 32 and 64 × 64.

Metric	Resolution	DFNO-2	DFNO-4	FFNO-32	FFNO-64
MSE	32 × 32	**0.0004**	**0.0012**	0.0101	0.0113
MSE	64 × 64	**0.0018**	**0.0007**	0.0136	0.0118
MAE	32 × 32	**0.0124**	**0.0208**	0.0677	0.0725
MAE	64 × 64	**0.0246**	**0.0168**	0.0788	0.0739



Epoch 600: Prediction

Epoch 600: Target

Epoch 600: Target - Prediction

Figure 2: This figure shows the downscaling performance of our DFNO model with softmax constraint layer on the PDE solution data. The DFNO model was trained with 2 times downscaling data, then evaluated at 1 time (row 1), 2 times (row 2), and 4 times (row 3) downscaling. Column 1 shows the outputs from our DFNO model; column 2 is the numerical solution ground truth; and the difference between truth and prediction is presented in column 3.

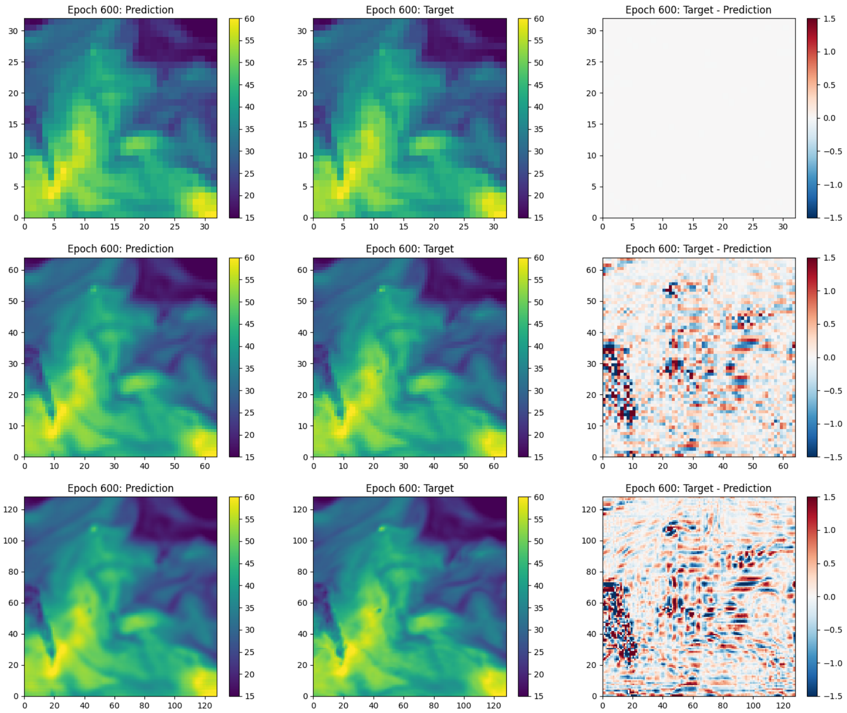


Figure 3: This figure shows the downscaling performance of our DFNO model with softmax constraint layer on ERA5 water content data. The DFNO model was trained with 2 times downscaling data, then evaluated at 1 time (row 1), 2 times (row 2), and 4 times (row 3) downscaling. Column 1 shows the outputs from our DFNO model; column 2 is the ground truth; and the difference between truth and prediction is presented in column 3.

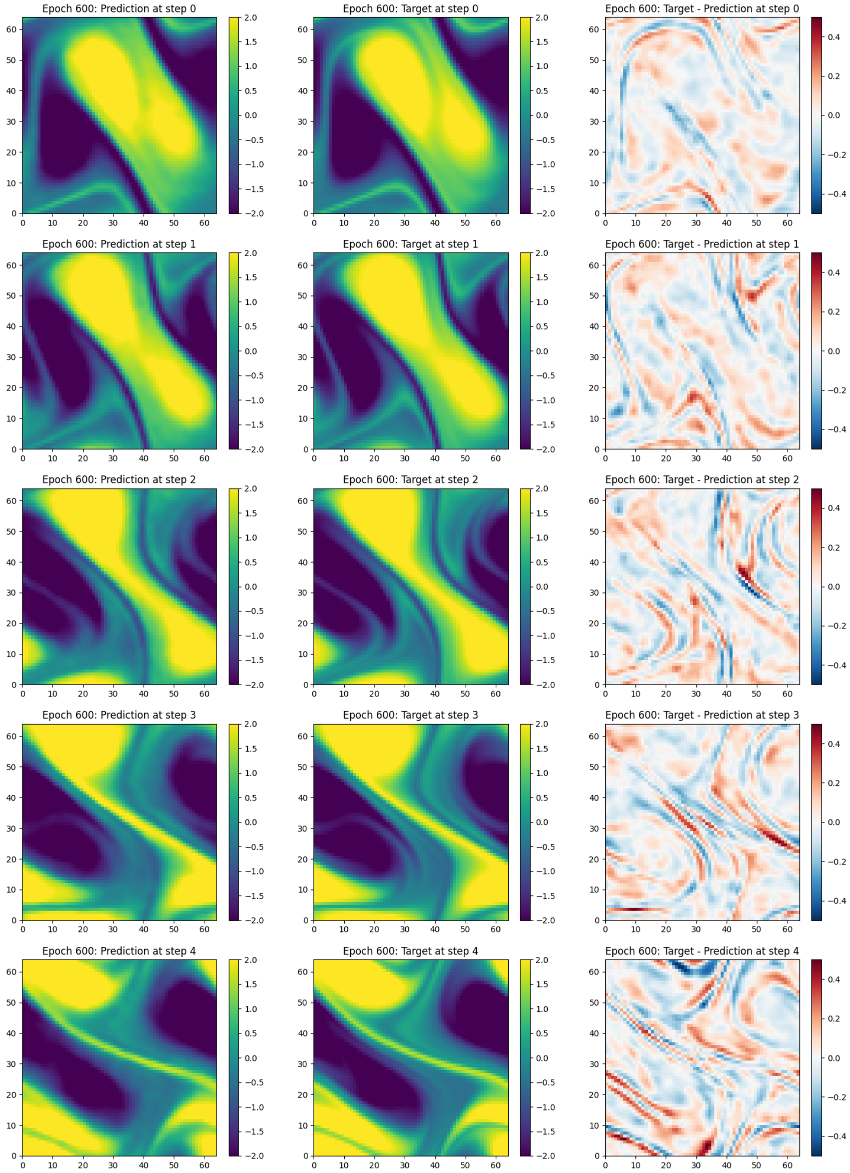


Figure 4: This figure shows Navier-Stokes equation solution (64×64) at five consecutive time steps (row 1 to row 5). The solution is generated by FFNO-64, a forward solution prediction model trained on a solution dataset of resolution 64×64. Column 1 shows FFNO-64 predicted solution; column 2 is the numerical solution ground truth; column 3 shows the difference between column 1 and column 2.

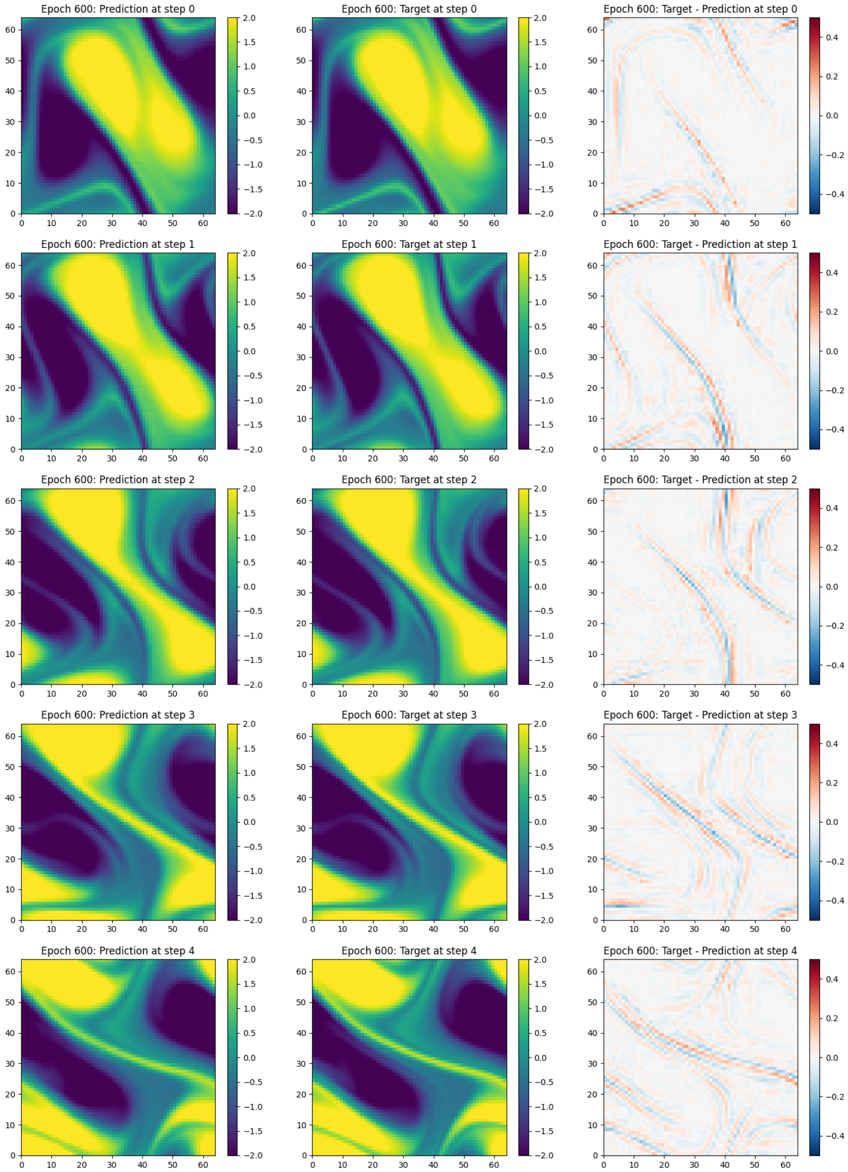


Figure 5: Similar to Figure 4 but the solution is generated by DFNO-2. It is a constrained DFNO model trained on solution downscaling data from 16×16 to 32×32. It performs zero-shot downscaling on a solution from 16×16 to 64×64.