Probabilistic neural operators

Christopher Bülte

(Ludwig-Maximilians-Universität München)

MIAPbP, September 10, 2025

Motivation

Many fundamental problems require to approximate some (nonlinear) operator $\mathcal{L}$:

- Inverse Problems $\rightarrow$ Inverse operator
- PDE $\rightarrow$ Solution operator
- Image denoising $\rightarrow$ Deconvolution operator

Motivation

Many fundamental problems require to approximate some (nonlinear) operator $\mathcal{L}$:

- Inverse Problems $\rightarrow$ Inverse operator
- PDE $\rightarrow$ Solution operator
- Image denoising $\rightarrow$ Deconvolution operator

Neural operators:

- Special type of neural network that approximates any nonlinear continuous operator
- Huge success in practical applications
  - PDE solving [Li et al., 2021]
  - Weather forecasting [Pathak et al., 2022, Bonev et al., 2023]
  - Scattering problem / Schrödinger operator [Mizera, 2023]

Motivation

Inherent probabilistic problems:

- Many problems do not have a deterministic or unique solution
- At the same time, uncertainty quantification might be of importance
  - Weather forecasting [Bülte et al., 2025]
  - Ocean dynamics [Chattopadhyay et al., 2024]
  - Climate modeling [Yang et al., 2023]

$\rightarrow$ Use neural operators in a *probabilistic* manner.

Overview

Structure:

1. Neural operators
2. Scoring rules for generative modeling
3. Scoring rules in infinite-dimensional spaces
4. Numerical results

(Fourier) neural operators

Neural operators

Preliminaries: Consider a bouned, open set $D \subset \mathbb{R}^d$ and separable Banach spaces (of functions) $\mathcal{A} = \mathcal{A}(D; \mathbb{R}^{d_a})$ and $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^{d_u})$. We are interested in some (nonlinear) operator

$$\mathcal{L}^\dagger : \mathcal{A} \to \mathcal{U}. \tag{1}$$

Neural operators

Preliminaries: Consider a bouned, open set $D \subset \mathbb{R}^d$ and separable Banach spaces (of functions) $\mathcal{A} = \mathcal{A}(D; \mathbb{R}^{d_a})$ and $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^{d_u})$. We are interested in some (nonlinear) operator

$$\mathcal{L}^\dagger : \mathcal{A} \to \mathcal{U}. \tag{1}$$

Neural operators: Given some observations $\{(a_j, u_j)\}_{j=1}^N$ neural operators aim to approximate $\mathcal{L}^\dagger$ via a parametric map

$$\mathcal{L}_\theta : \mathcal{A} \to \mathcal{U}, \quad \theta \in \Theta \subseteq \mathbb{R}^p, \tag{2}$$

by choosing $\theta^\dagger \in \Theta$ such that $\mathcal{L}_{\theta^\dagger} \approx \mathcal{L}^\dagger$.

Neural operators

Preliminaries: Consider a bouned, open set $D \subset \mathbb{R}^d$ and separable Banach spaces (of functions) $\mathcal{A} = \mathcal{A}(D; \mathbb{R}^{d_a})$ and $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^{d_u})$. We are interested in some (nonlinear) operator

$$\mathcal{L}^\dagger : \mathcal{A} \to \mathcal{U}. \tag{1}$$

Neural operators: Given some observations $\{(a_j, u_j)\}_{j=1}^N$ neural operators aim to approximate $\mathcal{L}^\dagger$ via a parametric map

$$\mathcal{L}_\theta : \mathcal{A} \to \mathcal{U}, \quad \theta \in \Theta \subseteq \mathbb{R}^p, \tag{2}$$

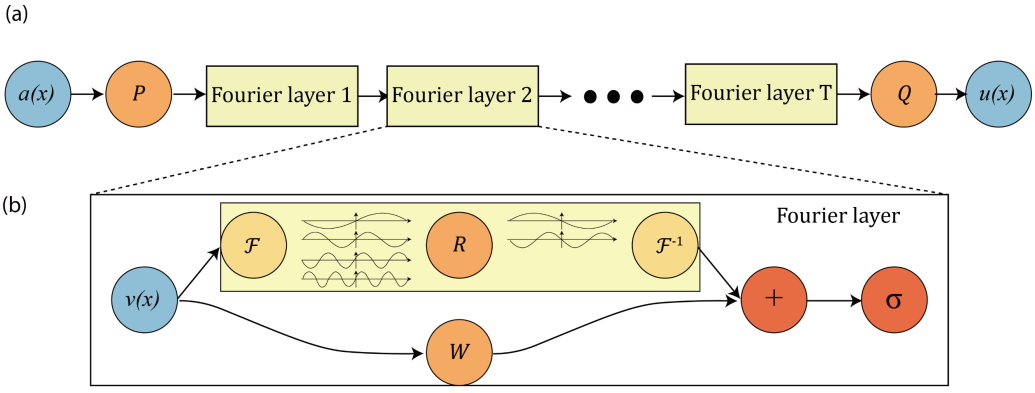
by choosing $\theta^\dagger \in \Theta$ such that $\mathcal{L}_{\theta^\dagger} \approx \mathcal{L}^\dagger$.

(Possible) Advantages

- Resolution-invariant
- More accurate than comparing DL methods
- Much faster than classical solvers

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Fourier neural operators



Fourier neural operator architecture [Li et al., 2021] with learned global (R) and local W features.

Fourier neural operators

Definition (Fourier neural operator)

Specific instance of neural operators, which specifies a function update in the ith layer as

$$\mathcal{L}_\phi v_i(x) = \sigma\left(\mathcal{F}^{-1}(R_{i,\phi} \cdot \mathcal{F}(v_i))(x) + W_{i,\phi} v_i(x)\right), \quad \forall x \in D, \tag{3}$$

where $W_{i,\phi} : \mathbb{R}^{d_v} \to \mathbb{R}^{d_v}$ is a linear transformation, $\sigma : \mathbb{R} \to \mathbb{R}$ is a pointwise nonlinearity, $\mathcal{F}$ is the Fourier transform and $R_\phi = \mathcal{F}(\kappa_\phi)$, with $\kappa_\phi : D \to \mathbb{R}^{d_v}$ is a function parametrized by neural network parameters ϕ.

The probabilistic framework

Definition (Probabilistic neural operator)

Assume we have observational data $\{(a_j, u_j)\}_{j=1}^N$ with $a_j \sim P_{\mathcal{A}}$ and $u_j \sim P^*_{\mathcal{U}}(\cdot \mid a_j)$. We consider a *probabilistic neural operator* as the map

$$\mathcal{L}_\theta : \mathcal{A} \times \mathcal{Z} \to \mathcal{U}, \quad \theta \in \Theta, \tag{4}$$

that aims to learn a distribution $P_\theta(\cdot \mid a_j) \coloneqq \mathcal{L}_\theta(a_j, \cdot)$ that fulfills $P_\theta(\cdot \mid a_j) \approx P^*_{\mathcal{U}}(\cdot \mid a_j)$.

The probabilistic framework

Definition (Probabilistic neural operator)

Assume we have observational data $\{(a_j, u_j)\}_{j=1}^N$ with $a_j \sim P_{\mathcal{A}}$ and $u_j \sim P^*_{\mathcal{U}}(\cdot \mid a_j)$. We consider a *probabilistic neural operator* as the map

$$\mathcal{L}_\theta : \mathcal{A} \times \mathcal{Z} \to \mathcal{U}, \quad \theta \in \Theta, \tag{4}$$

that aims to learn a distribution $P_\theta(\cdot \mid a_j) \coloneqq \mathcal{L}_\theta(a_j, \cdot)$ that fulfills $P_\theta(\cdot \mid a_j) \approx P^*_{\mathcal{U}}(\cdot \mid a_j)$.

- "Typical" generative setting: Samples are obtained via $z \sim P_{\mathcal{Z}}$
- Other approaches/definitions exist, for example, via (functional) Gaussian processes [Magnani et al., 2025]

Scoring rules for generative modeling

Proper scoring rules

Definition (Proper scoring rule)

Let $\mathcal{M}_1(\mathcal{X})$ denote the class of all probability measures on $\mathcal{X}$. For $\mathcal{P} \in \mathcal{M}_1(\mathcal{X})$, a *scoring rule* [Gneiting and Raftery, 2007] is a function $S : \mathcal{P} \times \mathcal{Y} \to \overline{\mathbb{R}}$, such that the integral

$$S(\mathbb{P}, \mathbb{Q}) := \mathbb{E}_{Y \sim \mathbb{Q}}[S(\mathbb{P}, Y)]$$

exists for all $\mathbb{P}, \mathbb{Q} \in \mathcal{P}$ [Gneiting and Raftery, 2007]. A scoring rule S is called *proper*, if

$$S(\mathbb{Q}, \mathbb{Q}) \leq S(\mathbb{P}, \mathbb{Q}), \quad \text{for all } \mathbb{P}, \mathbb{Q} \in \mathcal{P} \tag{5}$$

and *strictly proper* if (5) holds with equality if and only if $\mathbb{P} = \mathbb{Q}$.

Proper scoring rules

Definition (Proper scoring rule)

Let $\mathcal{M}_1(\mathcal{X})$ denote the class of all probability measures on $\mathcal{X}$. For $\mathcal{P} \in \mathcal{M}_1(\mathcal{X})$, a *scoring rule* [Gneiting and Raftery, 2007] is a function $S : \mathcal{P} \times \mathcal{Y} \to \overline{\mathbb{R}}$, such that the integral

$$S(\mathbb{P}, \mathbb{Q}) \coloneqq \mathbb{E}_{Y \sim \mathbb{Q}}[S(\mathbb{P}, Y)]$$

exists for all $\mathbb{P}, \mathbb{Q} \in \mathcal{P}$ [Gneiting and Raftery, 2007]. A scoring rule S is called *proper*, if

$$S(\mathbb{Q}, \mathbb{Q}) \leq S(\mathbb{P}, \mathbb{Q}), \quad \text{for all } \mathbb{P}, \mathbb{Q} \in \mathcal{P} \tag{5}$$

and *strictly proper* if (5) holds with equality if and only if $\mathbb{P} = \mathbb{Q}$.

- S measures the "distance" between an observation and a probability distribution
- Strictly proper $\rightarrow$ unique minimum
- Minimizing $S \rightarrow$ Minimizing the corresponding divergence

Proper scoring rules

Examples:

- $S_{\log} := -\log p(y)$ (NLL) $\rightarrow$ Strictly proper for $\mathcal{P}$ absolutely continuous
- $S_{\text{MSE}} := (\delta_{y^*} - y)^2 \rightarrow$ Strictly proper for $\mathcal{P}$ class of Dirac measures
- $S_{\text{Brier}} := (q - y)^2 \rightarrow$ Strictly proper for $\mathcal{P}$ all Bernoulli distributions
- $S_{\text{ES}} := \mathbb{E}_{X \sim P}[\|X - y\|] - \frac{1}{2}\mathbb{E}_{X, X' \sim P}[\|X - X'\|] \rightarrow$ strictly proper if expectations exist

Generative modeling with scoring rules

Recall

For $\mathcal{X} \subseteq \mathbb{R}^{d_x}, \mathcal{Y} \subseteq \mathbb{R}^{d_y}$, represent a conditional distribution $P_\theta(\cdot \mid x)$ via a map $h_\theta : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$. Samples can be obtained via $z \sim P_{\mathcal{Z}}$ and the corresponding pushforward measure.

Generative modeling with scoring rules

Recall

For $\mathcal{X} \subseteq \mathbb{R}^{d_x}, \mathcal{Y} \subseteq \mathbb{R}^{d_y}$, represent a conditional distribution $P_\theta(\cdot \mid x)$ via a map $h_\theta : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$. Samples can be obtained via $z \sim P_{\mathcal{Z}}$ and the corresponding pushforward measure.

Training objective [Pacchiardi et al., 2024]

Given data $(x_j, y_j)_{j=1}^N$ with prior $x_j \sim P_{\mathcal{X}}$ and $y_j \sim P^*_{\mathcal{Y}}(\cdot \mid x_j)$. Then the solution of the objective

$$\min_\theta \mathbb{E}_{X \sim P_{\mathcal{X}}} \mathbb{E}_{Y \sim P^*_{\mathcal{Y}}(\cdot \mid X)} S(P_\theta(\cdot \mid x), Y), \tag{6}$$

fulfills $P_\theta(\cdot \mid x) = P^*_{\mathcal{Y}}(\cdot \mid x), P_{\mathcal{X}}$-almost everywhere for a strictly proper scoring rule S.

Generative modeling with scoring rules

Estimator of objective

Actual training is possible via empirical (unbiased) estimators, which in the case of the energy score leads to

$$\operatorname{argmin}_\theta \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{M} \sum_{j=1}^{M} \|\hat{y}_i^j - y_i\|_2 - \frac{1}{2M(M-1)} \sum_{j,h=1, j \neq h}^{M} \|\hat{y}_i^j - \hat{y}_i^h\|_2 \right), \tag{7}$$

for $\hat{y}_i^j \sim P_\theta(\cdot \mid x_i), j = 1, \ldots, M$.

Generative modeling with scoring rules

Estimator of objective

Actual training is possible via empirical (unbiased) estimators, which in the case of the energy score leads to

$$\operatorname{argmin}_\theta \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{M} \sum_{j=1}^{M} \|\hat{y}_i^j - y_i\|_2 - \frac{1}{2M(M-1)} \sum_{j,h=1, j \neq h}^{M} \|\hat{y}_i^j - \hat{y}_i^h\|_2 \right), \tag{7}$$

for $\hat{y}_i^j \sim P_\theta(\cdot \mid x_i), j = 1, \ldots, M$.

Some results

- Successful application in weather modeling [Pacchiardi et al., 2024] or regression [Shen and Meinshausen, 2025]
- Extrapolation results and finite-sample convergence [Shen and Meinshausen, 2025]

Scoring rules in infinite dimensional spaces

Generative modeling

Definition (Probabilistic neural operator)

We consider a *probabilistic neural operator* as the map

$$\mathcal{L}_\theta : \mathcal{A} \times \mathcal{Z} \to \mathcal{U}, \quad \theta \in \Theta, \tag{8}$$

that aims to learn a distribution $P_\theta(\cdot \mid a_j) \coloneqq \mathcal{L}_\theta(a_j, \cdot)$ that fulfills $P_\theta(\cdot \mid a_j) \approx P^*_{\mathcal{U}}(\cdot \mid a_j)$.

Generative modeling

Definition (Probabilistic neural operator)

We consider a *probabilistic neural operator* as the map

$$\mathcal{L}_\theta : \mathcal{A} \times \mathcal{Z} \to \mathcal{U}, \quad \theta \in \Theta, \tag{8}$$

that aims to learn a distribution $P_\theta(\cdot \mid a_j) \coloneqq \mathcal{L}_\theta(a_j, \cdot)$ that fulfills $P_\theta(\cdot \mid a_j) \approx P^*_{\mathcal{U}}(\cdot \mid a_j)$.

Generic recipe

1. Specify (Fourier) neural operator architecture
2. Adjust model to generative setting, e.g., dropout, reparametrization, noise injection, etc.
3. Obtain M samples in forward pass
4. Train using the energy score S_{ES}

The energy score in separable Hilbert spaces

Theorem (B., Scholl, Kutyniok (2025))

Let $\mathcal{H}$ denote a separable Hilbert space and $x \in \mathcal{H}$. The energy score $\mathrm{ES} : \mathcal{M}_1^E(\mathcal{H}) \times \mathcal{H} \to \mathbb{R}$ defined as

$$\mathrm{ES}(P, x) := \mathbb{E}_P[\|X - x\|_{\mathcal{H}}] - \frac{1}{2}\mathbb{E}_P\left[\left\|X - X'\right\|_{\mathcal{H}}\right], \tag{9}$$

with $X, X' \overset{i.i.d}{\sim} P \in \mathcal{M}_1^E(\mathcal{H}) := \left\{P \in \mathcal{M}_1(\mathcal{H}) \mid \int_{\mathcal{H}} \|x\|_{\mathcal{H}} dP(x) < \infty\right\}$ is strictly proper relative to the class $\mathcal{M}_1^E(\mathcal{H})$.

Numerical experiments

Numerical experiments

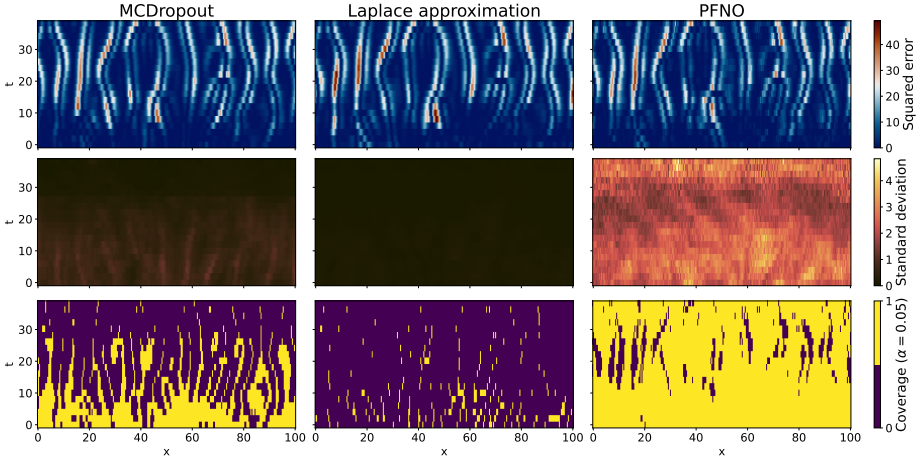
Setting

- Evaluate several PDES: Darcy Flow, KS, SWE, and Weather prediction task
- For generative modeling, use dropout and reparametrization model
- Compare to benchmark approaches

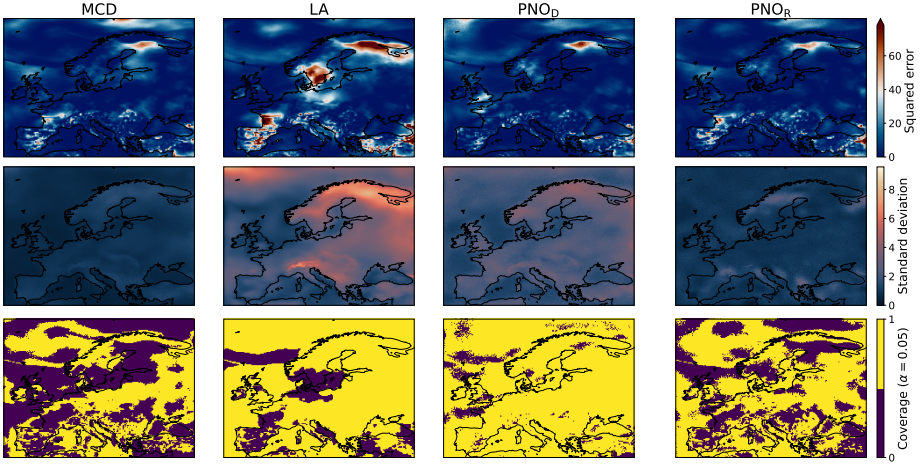
Darcy Flow

Model	Method	$\mathcal{L}_{L^2}$	ES	CRPS	NLL	$\mathcal{C}_{0.05}$	$\lvert\mathcal{C}_{0.05}\rvert$
FNO	DET	0.0972 (± 0.0076)	0.0972 (± 0.0076)	0.0783 (± 0.0062)	-	-	-
	PNO_D	**0.0772** (±0.0015)	**0.0569** (±0.0009)	**0.0462** (±0.0007)	**-1.3444** (±0.0355)	**0.9813** (±0.0032)	0.3682 (±0.0185)
	PNO_R	0.1031 (±0.0069)	0.0817 (±0.0056)	0.0707 (±0.0040)	685.15 (±913.21)	0.4702 (±0.0559)	0.1249 (±0.0191)
	MCD	0.1002 (±0.0088)	0.0830 (±0.0080)	0.0672 (±0.0066)	1.5973 (±0.8223)	0.5187 (±0.0608)	0.1203 (±0.0031)
	LA	0.0985 (±0.0074)	0.0767 (±0.0040)	0.0621 (±0.0033)	-0.8851 (±0.0531)	**0.9762** (±0.0085)	0.5713 (±0.0076)

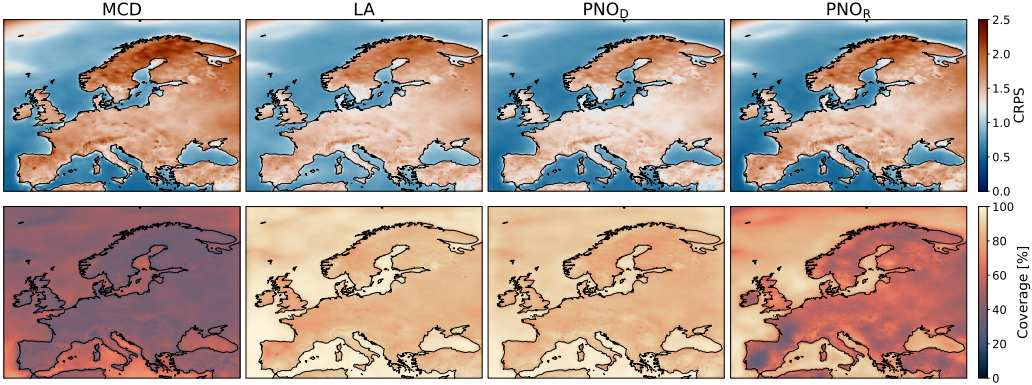
Kuramoto-Sivashinsky equation



Temperature forecasting



Temperature forecasting



Summary

Summary

- Extension of neural operators to a generative setting
- Theoretically justified training approach with loss function adapted to infinite dimensions
- Better calibration and prediction performance

Future work

Variational neural operators

- Design (conditional) variational autoencoder for inference in infinite dimensions
- Utilize ELBO in infinite dimensions or a Wasserstein-autoencoder with MMD objective
- Could allow for generation or inference for functional data

Future work

Variational neural operators

- Design (conditional) variational autoencoder for inference in infinite dimensions
- Utilize ELBO in infinite dimensions or a Wasserstein-autoencoder with MMD objective
- Could allow for generation or inference for functional data

Infinite dimensional diffusion models

- Recent work [Pidstrigach et al.,] consider diffusion processes in function spaces
- Combination with neural operators as denoiser network could lead to improvements in PDE trajectory prediction and similar tasks
- Theory in infinite dimensions puts restrictions on the noise process and the design is more complex

References I

Bonev, B., Kurth, T., Hundt, C., Pathak, J., Baust, M., Kashinath, K., and Anandkumar, A. (2023).
Spherical Fourier neural operators: Learning stable dynamics on the sphere.
In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, pages 2806–2823, Honolulu, Hawaii, USA. JMLR.org.

Bülte, C., Scholl, P., and Kutyniok, G. (2025).
Probabilistic neural operators for functional uncertainty quantification.

Chattopadhyay, A., Gray, M., Wu, T., Lowe, A. B., and He, R. (2024).
Oceannet: A principled neural operator-based digital twin for regional oceans.

Gneiting, T. and Raftery, A. E. (2007).
Strictly proper scoring rules, prediction, and estimation.
Journal of the American Statistical Association, 102(477):359–378.

References II

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2021).
Fourier Neural Operator for Parametric Partial Differential Equations.

Magnani, E., Pförtner, M., Weber, T., and Hennig, P. (2025).
Linearization turns neural operators into function-valued gaussian processes.

Mizera, S. (2023).
Scattering with neural operators.
Physical Review D, 108(10).

Pacchiardi, L., Adewoyin, R. A., Dueben, P., and Dutta, R. (2024).
Probabilistic forecasting with generative networks via scoring rule minimization.
Journal of Machine Learning Research, 25(45):1–64.

References III

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., Hassanzadeh, P., Kashinath, K., and Anandkumar, A. (2022).
FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators.

Pidstrigach, J., Marzouk, Y., Reich, S., and Wang, S.
Infinite-Dimensional Diffusion Models.

Shen, X. and Meinshausen, N. (2025).
Engression: Extrapolation through the lens of distributional regression.
Journal of the Royal Statistical Society Series B: Statistical Methodology, 87(3):653–677.

Yang, Q., Hernandez-Garcia, A., Harder, P., Ramesh, V., Sattegeri, P., Szwarcman, D., Watson, C. D., and Rolnick, D. (2023).
Fourier neural operators for arbitrary resolution climate data downscaling.