

---

# Operator learning for multi-patch domains

Massimiliano Ghiotto  
Departement of Mathematics, University of Pavia

*Joint work with:*

Carlo Marcati<sup>1</sup>    Giancarlo Sangalli<sup>1</sup>

---



# Table of Contents

- ▶ Fourier Neural Operator
- ▶ Multi-patch domains
- ▶ Numerical experiments



UNIVERSITÀ DI PAVIA



$$\mathcal{N}_\theta : \mathcal{A}(D, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_u}), \quad \text{with } D \subset \mathbb{R}^d, \quad \mathcal{N}_\theta := \mathcal{Q} \circ \mathcal{L}_L \circ \dots \circ \mathcal{L}_1 \circ \mathcal{P}.$$

1. **Lifting:** linear and local operator

$$\mathcal{P} : \mathcal{A}(D, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_v}), \quad \mathcal{P}(a)(x) = W_{\mathcal{P}} a(x) + b_{\mathcal{P}}, \quad W_{\mathcal{P}} \in \mathbb{R}^{d_v \times d_a}, b_{\mathcal{P}} \in \mathbb{R}^{d_v}$$

$$\mathcal{N}_\theta : \mathcal{A}(D, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_u}), \quad \text{with } D \subset \mathbb{R}^d, \quad \mathcal{N}_\theta := \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{P}.$$

1. **Lifting:** linear and local operator

$$\mathcal{P} : \mathcal{A}(D, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_v}), \quad \mathcal{P}(a)(x) = W_{\mathcal{P}} a(x) + b_{\mathcal{P}}, \quad W_{\mathcal{P}} \in \mathbb{R}^{d_v \times d_a}, b_{\mathcal{P}} \in \mathbb{R}^{d_v}$$

2. **Integral operator:** for  $t = 1, \dots, L$

$$\mathcal{L}_t : \mathcal{U}(D, \mathbb{R}^{d_v}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_v})$$

$$\mathcal{L}_t(v)(x) := \sigma \left( W_t v(x) + b_t + (\mathcal{K}_t(a, \theta)v)(x) \right)$$

with  $\mathcal{K}_t(a, \theta)$  linear and non-local operator.

$$\mathcal{N}_\theta : \mathcal{A}(D, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_u}), \quad \text{with } D \subset \mathbb{R}^d, \quad \mathcal{N}_\theta := \mathcal{Q} \circ \mathcal{L}_L \circ \dots \circ \mathcal{L}_1 \circ \mathcal{P}.$$

1. **Lifting:** linear and local operator

$$\mathcal{P} : \mathcal{A}(D, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_v}), \quad \mathcal{P}(a)(x) = W_{\mathcal{P}} a(x) + b_{\mathcal{P}}, \quad W_{\mathcal{P}} \in \mathbb{R}^{d_v \times d_a}, b_{\mathcal{P}} \in \mathbb{R}^{d_v}$$

2. **Integral operator:** for  $t = 1, \dots, L$

$$\mathcal{L}_t : \mathcal{U}(D, \mathbb{R}^{d_v}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_v})$$

$$\mathcal{L}_t(v)(x) := \sigma \left( W_t v(x) + b_t + (\mathcal{K}_t(a, \theta)v)(x) \right)$$

with  $\mathcal{K}_t(a, \theta)$  linear and non-local operator.

3. **Projection:** linear and local operator

$$\mathcal{Q} : \mathcal{U}(D, \mathbb{R}^{d_v}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_u}), \quad \mathcal{Q}(v)(x) = W_{\mathcal{Q}} v(x) + b_{\mathcal{Q}}, \quad W_{\mathcal{Q}} \in \mathbb{R}^{d_u \times d_v}, b_{\mathcal{Q}} \in \mathbb{R}^{d_u}.$$

# Definition of Fourier Neural Operator

## 1 Fourier Neural Operator

---

For defining the Fourier Neural Operator (FNO) we make the following assumptions

$$(\mathcal{K}_t(\theta_t, a)v)(x) = (\mathcal{K}_t(\theta_t)v)(x) = \int_{\mathbb{T}^d} \kappa_{t,\theta_t}(x, y)v(y) dy = \int_{\mathbb{T}^d} \kappa_{t,\theta_t}(x-y)v(y) dy = (\kappa_{t,\theta_t} * v)(x).$$

For defining the Fourier Neural Operator (FNO) we make the following assumptions

$$(\mathcal{K}_t(\theta_t, a)v)(x) = (\mathcal{K}_t(\theta_t)v)(x) = \int_{\mathbb{T}^d} \kappa_{t,\theta_t}(x, y)v(y) dy = \int_{\mathbb{T}^d} \kappa_{t,\theta_t}(x-y)v(y) dy = (\kappa_{t,\theta_t} * v)(x).$$

Using the convolution theorem we have

$$(\kappa_{t,\theta_t} * v)(x) = \mathcal{F}^{-1}(\mathcal{F}(\kappa_{t,\theta_t})(k) \cdot \mathcal{F}(v)(k))(x),$$

and parameterizing  $\mathcal{F}(\kappa_{t,\theta_t})$  with the parameters  $R_{\theta_t}(k) \in \mathbb{C}^{d_v \times d_v} \forall k \in \mathbb{Z}^d$ , we have

$$(\mathcal{K}_t(\theta_t)v)(x) = \mathcal{F}^{-1}(R_{\theta_t}(k) \cdot \mathcal{F}(v)(k))(x).$$



### Universal approximation theorem

Given  $s, s' \geq 0$  and

$$\mathcal{G} : H^s(\mathbb{T}^d, \mathbb{R}^{d_a}) \rightarrow H^{s'}(\mathbb{T}^d, \mathbb{R}^{d_u})$$

continuous operator. Given  $K \subset H^s(\mathbb{T}^d, \mathbb{R}^{d_a})$  a compact subset and  $\sigma \in \mathbb{C}^\infty(\mathbb{R})$  non-linear and globally Lipschitz activation function. Then, for all  $\varepsilon > 0$ , exists a Fourier Neural Operator

$$\mathcal{N} : H^s(\mathbb{T}^d, \mathbb{R}^{d_a}) \rightarrow H^{s'}(\mathbb{T}^d, \mathbb{R}^{d_u})$$

such that:

$$\sup_{a \in K} \|\mathcal{G}(a) - \mathcal{N}(a)\|_{H^{s'}} \leq \varepsilon.$$

<sup>1</sup>N. Kovachki, S. Lanthaler, S. Mishra, "On universal approximation and error bounds for fourier neural operators"



Pseudo Spectral Fourier Neural Operator ( $\psi$ -FNO) is a map

$$\mathcal{N}^* : \mathcal{A}(\mathbb{T}^d, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(\mathbb{T}^d, \mathbb{R}^{d_u}), \quad a \mapsto \mathcal{N}^*(a),$$

defined by

$$\mathcal{N}^*(a) = \mathcal{Q} \circ I_N \circ \mathcal{L}_L \circ I_N \circ \cdots \circ \mathcal{L}_1 \circ I_N \circ \mathcal{R}(a),$$

where  $I_N$  denotes the pseudo-spectral projection on the Fourier polynomials of degree  $N$

$$I_N : C(\mathbb{T}^d) \rightarrow L_N^2(\mathbb{T}^d), \quad u \mapsto I_N u.$$



## 1 Fourier Neural Operator

Pseudo Spectral Fourier Neural Operator ( $\psi$ -FNO) is a map

$$\mathcal{N}^* : \mathcal{A}(\mathbb{T}^d, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(\mathbb{T}^d, \mathbb{R}^{d_u}), \quad a \mapsto \mathcal{N}^*(a),$$

defined by

$$\mathcal{N}^*(a) = \mathcal{Q} \circ I_N \circ \mathcal{L}_L \circ I_N \circ \cdots \circ \mathcal{L}_1 \circ I_N \circ \mathcal{R}(a),$$

where  $I_N$  denotes the pseudo-spectral projection on the Fourier polynomials of degree  $N$

$$I_N : C(\mathbb{T}^d) \rightarrow L_N^2(\mathbb{T}^d), \quad u \mapsto I_N u.$$

So a  $\psi$ -FNO can be identified with a finite-dimensional map

$$\tilde{\mathcal{N}}^* : \mathbb{R}^{d_a \times \mathcal{I}_N} \rightarrow \mathbb{R}^{d_u \times \mathcal{I}_N}, \quad \tilde{\mathcal{N}}^*(a)_j = \mathcal{N}^*(a)(x_j) \quad \forall j \in \mathcal{I}_N,$$

where the input  $a = \{a_j\}_{j \in \mathcal{I}_N}$ ,  $a_j = a(x_j)$  and  $\mathcal{I}_N = \{1, \dots, 2N\}^d$ .

# Universal approximation for $\psi$ -FNOs<sup>2</sup>

## 1 Fourier Neural Operator



### Universal approximation for $\psi$ -FNOs

Given  $s > d/2$ ,  $s' \geq 0$  and

$$\mathcal{G} : H^s(\mathbb{T}^d, \mathbb{R}^{d_a}) \rightarrow H^{s'}(\mathbb{T}^d, \mathbb{R}^{d_u})$$

continuous operator. Given  $K \subset H^s(\mathbb{T}^d, \mathbb{R}^{d_a})$  a compact set and  $\sigma \in \mathbb{C}^\infty(\mathbb{R})$  a non-linear and globally Lipschitz continuous function. Then, for any  $\varepsilon > 0$ , exists an  $N \in \mathbb{N}$  such that the  $\psi$ -FNO

$$\mathcal{N}^* : L_N^2(\mathbb{T}^d, \mathbb{R}^{d_a}) \rightarrow L_N^2(\mathbb{T}^d, \mathbb{R}^{d_u})$$

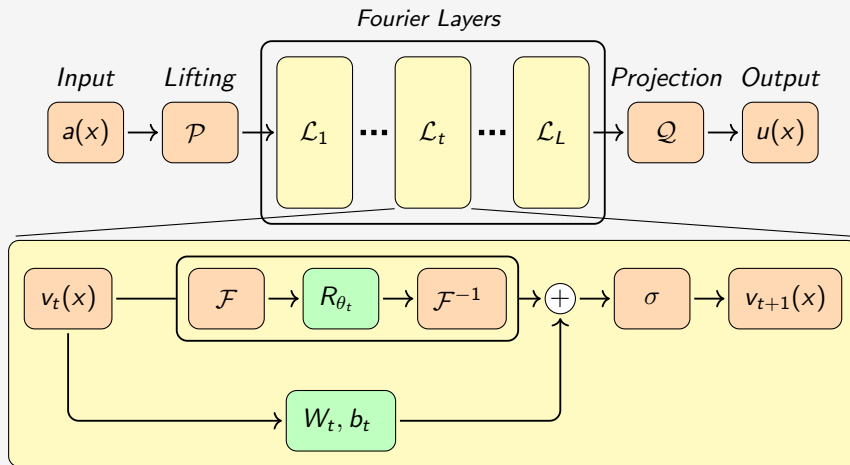
satisfy:

$$\sup_{a \in K} \|\mathcal{G}(a) - \mathcal{N}^*(a)\|_{H^{s'}} \leq \varepsilon.$$

<sup>2</sup>N. Kovachki, S. Lanthaler, S. Mishra, "On universal approximation and error bounds for fourier neural operators"

# Representation of $\psi$ -FNO

## 1 Fourier Neural Operator



# Table of Contents

- ▶ Fourier Neural Operator
- ▶ Multi-patch domains
- ▶ Numerical experiments



UNIVERSITÀ DI PAVIA

FNOs are limited to rectangular domains. To extend the operator to irregular domains, we can extend the domain to a larger rectangular domain and training the FNO to approximate the solution only on the original domain.

### Periodic extension operator

Let  $\Omega \subset \mathbb{R}^d$  be a bounded and Lipschitz domain. There exists a continuous linear operator  $\mathcal{E} : W^{m,p}(\Omega) \rightarrow W^{m,p}(B)$  for any  $m \geq 0$  and  $1 \leq p < \infty$ , where  $B$  is a hyper-cube containing  $\Omega$ , such that, for any  $u \in W^{m,p}(\Omega)$ :

- $\mathcal{E}(u)|_{\Omega} = u$ ,
- $\mathcal{E}(u)$  is periodic on  $B$ , including its derivatives.

FNOs are limited to rectangular domains. To extend the operator to irregular domains, we can extend the domain to a larger rectangular domain and training the FNO to approximate the solution only on the original domain.

### Periodic extension operator

Let  $\Omega \subset \mathbb{R}^d$  be a bounded and Lipschitz domain. There exists a continuous linear operator  $\mathcal{E} : W^{m,p}(\Omega) \rightarrow W^{m,p}(B)$  for any  $m \geq 0$  and  $1 \leq p < \infty$ , where  $B$  is a hyper-cube containing  $\Omega$ , such that, for any  $u \in W^{m,p}(\Omega)$ :

- $\mathcal{E}(u)|_{\Omega} = u$ ,
- $\mathcal{E}(u)$  is periodic on  $B$ , including its derivatives.

But this can be computationally expensive.

Let  $D \subset \mathbb{R}^d$  be a bounded domain. The Chebyshev Neural Operator (CNO) is a map

$$\mathcal{N}_\theta : \mathcal{A}(D, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_u}),$$

defined by

$$\mathcal{N}_\theta(a) = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{P}(a),$$

where is all defined as the FNO but in the integral operator we use the Chebyshev transform and anti-transform instead of the Fourier transform, i.e.

$$(\mathcal{K}_t(\theta_t)v)(x) = \mathcal{C}^{-1}(R_{\theta_t}(k) \cdot \mathcal{C}(v)(k))(x), \quad R_{\theta_t}(k) \in \mathbb{R}^{d_v \times d_v} \quad \forall k \in \mathbb{N}^d.$$



Let  $D \subset \mathbb{R}^d$  be a bounded domain. The Chebyshev Neural Operator (CNO) is a map

$$\mathcal{N}_\theta : \mathcal{A}(D, \mathbb{R}^{d_a}) \rightarrow \mathcal{U}(D, \mathbb{R}^{d_u}),$$

defined by

$$\mathcal{N}_\theta(a) = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{P}(a),$$

where is all defined as the FNO but in the integral operator we use the Chebyshev transform and anti-transform instead of the Fourier transform, i.e.

$$(\mathcal{K}_t(\theta_t)v)(x) = \mathcal{C}^{-1}(R_{\theta_t}(k) \cdot \mathcal{C}(v)(k))(x), \quad R_{\theta_t}(k) \in \mathbb{R}^{d_v \times d_v} \quad \forall k \in \mathbb{N}^d.$$

At the discrete level, we can define the Pseudo Spectral Chebyshev Neural Operator ( $\psi$ -CNO) as before, with the difference that the grid  $\mathcal{I}_N$  is defined on a Chebyshev grid and not on an uniform one.

# Boundary adapted Chebyshev basis

## 2 Multi-patch domains



The one dimensional Chebyshev polynomials are defined on the interval  $[-1, 1]$ , as  $T_n(\cos(\theta)) = \cos(n\theta)$ ,  $n \in \mathbb{N}$  with the property that  $T_n(1) = 1$  and  $T_n(-1) = (-1)^n$ . We define the boundary adapted Chebyshev basis as  $B_0(x) = \frac{1}{2}(-x + 1)$ ,  $B_1(x) = \frac{1}{2}(x + 1)$  and for  $n \geq 2$

$$B_n(x) = \begin{cases} T_n(x) - T_0(x) & n \text{ is even,} \\ T_n(x) - T_1(x) & n \text{ is odd.} \end{cases}$$

The one dimensional Chebyshev polynomials are defined on the interval  $[-1, 1]$ , as  $T_n(\cos(\theta)) = \cos(n\theta)$ ,  $n \in \mathbb{N}$  with the property that  $T_n(1) = 1$  and  $T_n(-1) = (-1)^n$ . We define the boundary adapted Chebyshev basis as  $B_0(x) = \frac{1}{2}(-x + 1)$ ,  $B_1(x) = \frac{1}{2}(x + 1)$  and for  $n \geq 2$

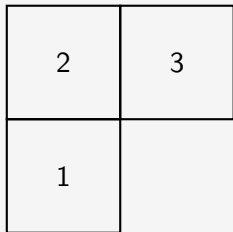
$$B_n(x) = \begin{cases} T_n(x) - T_0(x) & n \text{ is even,} \\ T_n(x) - T_1(x) & n \text{ is odd.} \end{cases}$$

For two dimensional Chebyshev polynomials we define  $B_{n_1, n_2}(x, y) = B_{n_1}(x)B_{n_2}(y)$ ,  $0 \leq n_1, n_2 \leq n$ . Considering the square domain  $D = [-1, 1]^2$  we notice that there is only a function not equal to zero in each vertex of the square and there are only  $n$  functions that are not zero on each edge,  $B_{i,j}(x, y)$   $2 \leq i, j \leq n$  are zero on the boundary.

If we have the domain  $D$  divided in  $N$  patches  $D_\ell$  with  $D = \bigcup_{\ell=1}^N D_\ell$  with  $D_i \cap D_j = \emptyset$  or  $D_i \cap D_j = \Gamma_{ij}$  for  $i \neq j$ .

We consider the L-domain

$$D = [-1, 1]^2 \setminus ([0, 1] \times [-1, 0])$$



divided in  $N = 3$  patches  $D_1, D_2, D_3$ . We define the boundary adapted Chebyshev basis on each patch  $D_\ell$  and denotes with  $B_{i,j}^\ell$  with  $k = 1, 2, 3$  and  $0 \leq i, j \leq n$ . To ensure the continuity on the interface  $\Gamma_{1,2}$  and  $\Gamma_{2,3}$  we have to impose that

$$b_{i,1}^0 = b_{i,0}^1, \quad b_{0,i}^2 = b_{1,i}^1, \quad \forall i = 1, \dots, n.$$

The multi-patch neural operator is defined as the Chebyshev neural operator on each patch  $D_\ell$  and the continuity on the interface is imposed using the boundary adapted Chebyshev basis in the following way:

- compute the Chebyshev discrete transform of the function  $v$  on each patch  $D_k$ ,  $(Cv^\ell)(k)$  with  $\ell = 1, 2, 3$  and  $0 \leq i, j \leq n$ ,
- multiply with the neural operator parameters  $R_{\theta_t}(k) \cdot (Cv^\ell)(k)$ ,
- compute the boundary adapted Chebyshev coefficients on each patch  $b^\ell(k)$ ,
- impose the continuity on the interface  $\Gamma_{1,2}$  and  $\Gamma_{2,3}$ ,

$$\text{update } b_{i,1}^0, b_{i,0}^1 \text{ with } \frac{b_{i,1}^0 + b_{i,0}^1}{2}, \quad \text{update } b_{0,i}^2, b_{1,i}^1 \text{ with } \frac{b_{0,i}^2 + b_{1,i}^1}{2}, \quad \forall i = 1, \dots, n.$$

- go back to the Chebyshev coefficients on each patch,
- compute the Chebyshev inverse transform on each patch.

# Table of Contents

- ▶ Fourier Neural Operator
- ▶ Multi-patch domains
- ▶ Numerical experiments



UNIVERSITÀ DI PAVIA



For our numerical experiments we consider the Darcy problem:

$$\begin{cases} -\nabla(a \cdot \nabla u) = f, & \text{in } D \\ u = 0, & \text{on } \partial D \end{cases}$$

with the L-domain  $D = [-1, 1]^2 \setminus ((0, 1) \times (-1, 0))$ , diffusion coefficient  $a \in \mathcal{A} = L^\infty(D, \mathbb{R}^+)$ , and  $f \equiv 1$ . With this settings exists a unique solution  $u \in H_0^1(D, \mathbb{R})$  and so we can define the solution operator

$$\mathcal{G} : L^\infty(D, \mathbb{R}^+) \rightarrow H_0^1(D, \mathbb{R}), \quad \mathcal{G} : a \mapsto u.$$

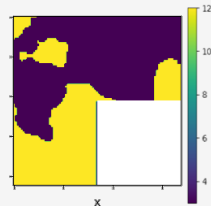


We have to generate the dataset  $\{a^{(i)}, u^{(i)}\}_{i=1}^N$ .



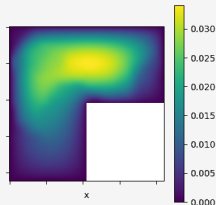
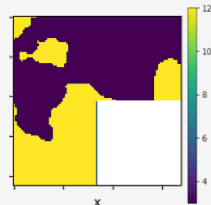
We have to generate the dataset  $\{a^{(i)}, u^{(i)}\}_{i=1}^N$ .

For the input we use the push forward of a proper gaussian random fields  $a^{(i)} \sim \mu = T_{\#}N(0, C)$  i.i.d. where  $C = -(\Delta + 9I)^{-2}$  and  $T : \mathbb{R} \rightarrow \mathbb{R}$  such that  $T$  gives 12 to the positive values and 3 to the negative ones.



We have to generate the dataset  $\{a^{(i)}, u^{(i)}\}_{i=1}^N$ .

For the input we use the push forward of a proper gaussian random fields  $a^{(i)} \sim \mu = T_{\#}N(0, C)$  i.i.d. where  $C = -(\Delta + 9I)^{-2}$  and  $T : \mathbb{R} \rightarrow \mathbb{R}$  such that  $T$  gives 12 to the positive values and 3 to the negative ones.



For the output we approximate the solution of the Darcy problem solution  $u^{(i)} = \mathcal{G}(a^{(i)})$  using the isogeometric geoPDEs library.



We define the relative  $L^2$  error as

$$\left\| \frac{\mathcal{G} - \mathcal{N}_\theta^*}{\mathcal{G}} \right\|_{L_\mu^2(L^\infty, L^2)} = \mathbb{E}_{a \sim \mu} \frac{\|\mathcal{G}(a) - \mathcal{N}_\theta^*(a)\|_{L^2(D)}^2}{\|\mathcal{G}(a)\|_{L^2(D)}^2}$$

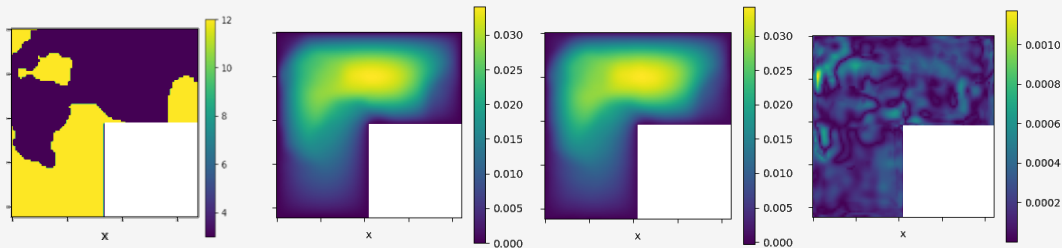
that can be approximated by

$$\left\| \frac{\mathcal{G} - \mathcal{N}_\theta^*}{\mathcal{G}} \right\|_{L_\mu^2(L^\infty, L^2)} \approx \frac{1}{N} \sum_{i=1}^N \left( \frac{\sum_{k=1}^M |u^{(i)}(x_k) - \mathcal{N}_\theta^*(a^{(i)})(x_k)|^2}{\sum_{k=1}^M |u^{(i)}(x_k)|^2} \right).$$

Where  $\{x_k\}_{k=1}^M \subset D$  is the set of the discretization points. That is a uniform (or Chebyshev) grid of 42 points per direction in every square patches.

# Fourier continuation numerical experiments

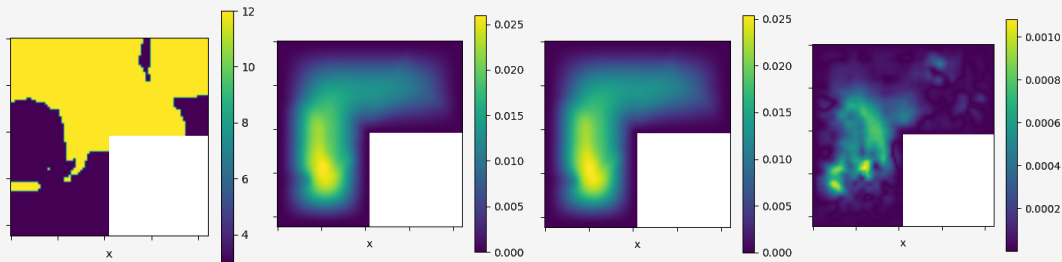
## 3 Numerical experiments



rel. train error $L^2$	rel. test error $L^2$	modes	parameters	epochs
0.0182	0.0229	12	2 363 681	500

# Multi-patch neural operator numerical experiments

## 3 Numerical experiments



rel. train error $L^2$	rel. test error $L^2$	modes	parameters	epochs
0.0115	0.0212	24	2372233	500



### Achievements:

- We have constructed a multi-patch neural operator for the Darcy problem on the L-domain.
- The neural operator is easily parallelizable between the patches.

### Achievements:

- We have constructed a multi-patch neural operator for the Darcy problem on the L-domain.
- The neural operator is easily parallelizable between the patches.

### Future works:

- Analyze if an universal approximation result can be obtained for the multi-patch neural operator.
- Use the multi-patch neural operator to approximate the solution of different PDEs.
- Extend the multi-patch neural operator to more complex geometries, with patches that are isomorphic to the square.

**Operator learning for multi-patch domains**

**Thank you for your attention!**



UNIVERSITÀ DI PAVIA





### Teorema

Given  $s > d/2$ ,  $\lambda \in (0, 1)$  and we consider the solution operator for the Darcy problem

$$\mathcal{G} : \mathcal{A}_\lambda^s(\mathbb{T}^d) \rightarrow H^1(\mathbb{T}^d).$$

Given  $\sigma \in C^3(\mathbb{R})$  non-polynomial, for any  $N \in \mathbb{N}$  exists  $C > 0$  and a  $\psi$ -FNO

$$\mathcal{N}^* : \mathcal{A}_\lambda^s(\mathbb{T}^d) \rightarrow H^1(\mathbb{T}^d)$$

such that

$$\sup_{a \in \mathcal{A}_\lambda^s} \|\mathcal{G}(a) - \mathcal{N}^*(a)\|_{H^1(\mathbb{T}^d)} \leq CN^{-k}$$

and  $\text{depth}(\mathcal{N}^*) \leq C \log(N)$ ,  $\text{lift}(\mathcal{N}^*) \leq C$ ,  $\text{size}(\mathcal{N}^*) \lesssim N^d \log(N)$ .

- Given  $v \in L^2(\mathbb{T}^d)$ , the Fourier transform is defined as

$$\begin{aligned}\mathcal{F} : L^2(\mathbb{T}^d, \mathbb{C}^n) &\rightarrow \ell^2(\mathbb{Z}^d, \mathbb{C}^n) \\ v &\mapsto \mathcal{F}(v)\end{aligned}$$

$$\mathcal{F}(v)(k) := \frac{1}{(2\pi)^d} \int_{\mathbb{T}^d} v(x) e^{-i\langle k, x \rangle} dx, \quad \forall k \in \mathbb{Z}^d.$$

- Given  $\hat{v} = \{\hat{v}_k\}_{k \in \mathbb{Z}^d} \in \ell^2(\mathbb{Z}^d, \mathbb{C}^n)$ , the anti-Fourier transform is defined as

$$\begin{aligned}\mathcal{F}^{-1} : \ell^2(\mathbb{Z}^d, \mathbb{C}^n) &\rightarrow L^2(D, \mathbb{C}^n) \\ \hat{v} &\mapsto \mathcal{F}^{-1}(\hat{v})\end{aligned}$$

$$(\mathcal{F}^{-1}\hat{v})(x) = \sum_{k \in \mathbb{Z}^d} \hat{v}_k e^{i\langle k, x \rangle} \quad \forall x \in D.$$

We choose  $N \in \mathbb{N}$  and fix an uniform grid  $\{x_j\}_{j \in \mathcal{I}_N}$  with  $x_j = (2\pi j)/(2N+1) \in \mathbb{T}^d$ ,  $j \in \mathcal{I}_N = \{0, \dots, 2N\}^d$ , finally we define  $\mathcal{K}_N := \{k \in \mathbb{Z}^d : |k|_\infty \leq N\}$ . We define the discrete Fourier transform as

$$\mathcal{F}_N : \mathbb{R}^{\mathcal{I}_N} \rightarrow \mathbb{C}^{\mathcal{K}_N}$$
$$\mathcal{F}_N(v)(k) := \frac{1}{(2N+1)^d} \sum_{j \in \mathcal{I}_N} v(x_j) e^{-2\pi i \langle j, k \rangle / N}, \quad \forall k \in \mathcal{K}_N,$$

and we define the discrete version of the anti-Fourier transform as

$$\mathcal{F}_N^{-1} : \mathbb{C}^{\mathcal{K}_N} \rightarrow \mathbb{R}^{\mathcal{I}_N}$$
$$\mathcal{F}_N^{-1}(\hat{v})(j) := \sum_{k \in \mathcal{K}_N} \hat{v}_k e^{2\pi i \langle j, k \rangle / N}, \quad \forall j \in \mathcal{I}_N.$$

# Sketch of the proof of universal approximation theorem for FNOs

## 3 Numerical experiments

---

- The definition of the projection on the trigonometric polynomial is

$$P_N : L^2(\mathbb{T}^d) \rightarrow L_N^2(\mathbb{T}^d),$$
$$P_N \left( \sum_{k \in \mathbb{Z}^d} c_k e^{i\langle x, k \rangle} \right) = \sum_{|k|_\infty \leq N} c_k e^{i\langle x, k \rangle}, \quad \forall (c_k)_{k \in \mathbb{Z}^d} \in \ell^2(\mathbb{Z}^d).$$

- If the universal approximation theorem holds for  $s' = 0$  then it holds for any value of  $s' \geq 0$ .
- We fix  $s' = 0$

$$\mathcal{G}_N : H^s(\mathbb{T}^d, \mathbb{R}^{d_a}) \rightarrow L^2(\mathbb{T}^d, \mathbb{R}^{d_u}), \quad \mathcal{G}_N(a) := P_N \mathcal{G}(P_N a),$$

holds that  $\forall \varepsilon > 0, \exists N \in \mathbb{N}$  such that

$$\|\mathcal{G}(a) - \mathcal{G}_N(a)\|_{L^2} \leq \varepsilon, \quad \forall a \in K.$$

# Sketch of the proof of universal approximation theorem for FNOs

## 3 Numerical experiments

---



Define the operator

$$\hat{\mathcal{G}}_N : \mathbb{C}^{\mathcal{K}_N} \rightarrow \mathbb{C}^{\mathcal{K}_N}, \quad \hat{\mathcal{G}}_N(\hat{a}_k) := \mathcal{F}_N(\mathcal{G}_N(\text{Re}(\mathcal{F}_N^{-1}(\hat{a}_k)))),$$

holds the following identity

$$\mathcal{G}_N(a) = \mathcal{F}_N^{-1} \circ \hat{\mathcal{G}}_N \circ \mathcal{F}_N(P_N a),$$

for any function  $a \in L^2(\mathbb{T}^d, \mathbb{R}^{d_a})$ . So we reconduce to demonstrate that FNO can approximate

$$\mathcal{F}_N^{-1}, \hat{\mathcal{G}}_N, \mathcal{F}_N(P_N a).$$



Let  $d \in \mathbb{N}$  and  $L \in \mathbb{N}$  with  $L \geq 2$  and  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  a non-linear activation function. Let  $A_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ ,  $b_\ell \in \mathbb{R}^{n_\ell}$  with  $n_\ell \in \mathbb{N}$  for  $\ell = 1, \dots, L$  and  $n_0 = d$ . We call multilayer perceptron (MLP) the function defined as

$$\begin{cases} x_L = A_L x_{L-1} + b_L \\ x_\ell = \sigma(A_\ell x_{\ell-1} + b_\ell) \end{cases},$$

where  $x_0$  is the input and  $x_L$  is the output of the function.



Suppose that  $\sigma \in TW$ ,  $X$  is a Banach space,  $K_1 \subset X$ ,  $K_2 \subset \mathbb{R}^d$  are two compact sets in  $X$  and  $\mathbb{R}^d$  respectively, and  $V$  is a compact set in  $C(K_1)$ . Let  $G$  a nonlinear continuous operator which maps  $V$  into  $C(K_2)$ , then for any  $\varepsilon > 0$ , there are a positive integers  $n, p, m$ ; real constants  $c_i^k, \theta_i^k, \xi_{ij}^k, \zeta^k \in \mathbb{R}$ , points  $w^k \in \mathbb{R}^d$  and  $x_j \in K_1$ , with  $i = 1, \dots, n, j = 1, \dots, m$  and  $k = 1, \dots, p$ , such that

$$\left| G(u)(y) - \sum_{k=1}^p \sum_{i=1}^n c_i^k \sigma \left( \sum_{j=1}^m \xi_{ij}^k u(x_j) + \theta_i^k \right) \sigma(w^k \cdot y + \zeta^k) \right| < \varepsilon$$

holds for all  $u \in V$  and  $y \in K_2$ .