

# SciML 2: Neural operators

E. Franck

Master CSMI, Strasbourg University

01/09/2025

# 1. Operator learning: Euclidian spectral approach

# Rappel I

**Definition** (Nonlocal parametric operator): On se donne une fonction  $\mathbf{v}_1 : \mathbb{R}^d \rightarrow \mathbb{R}^{\text{in}}$  et  $\mathbf{v}_{1+1} : \mathbb{R}^d \rightarrow \mathbb{R}^{\text{out}}$ . On appelle une couche non-locale l'opérateur

$$\mathbf{v}_{1+1}(\mathbf{x}) = \mathcal{B}_{\theta}(\mathbf{v})(\mathbf{x}) = \int_{\mathcal{D}} k_{\theta}(\mathbf{x}, \mathbf{y}, \mathbf{v}_1(\mathbf{x}), \mathbf{v}_1(\mathbf{y})) \mathbf{v}_1(\mathbf{y}) d\mathbf{y}$$

ou  $K_{\theta}$  le noyau sera la partie apprenable. Dans la majorité des cas l'opérateur est linéaire pour généraliser les couches affines des réseaux classiques.

**Definition** (Local parametric operator): On se donne une fonction  $\mathbf{v}_1 : \mathbb{R}^d \rightarrow \mathbb{R}^{\text{in}}$  et  $\mathbf{v}_{1+1} : \mathbb{R}^d \rightarrow \mathbb{R}^{\text{out}}$ . On appelle une couche non-locale l'opérateur

$$\mathbf{v}_{1+1}(\mathbf{x}) = \text{nn}_{\theta}(\mathbf{v}_1(\mathbf{x})), \forall \mathbf{x} \in \mathbb{R}^{\text{in}}$$

avec  $\text{nn}_{\theta}$  un réseau de neurones ou une couche paramétrée par  $\theta$ .

## Rappel II

**Definition** (Opérateur paramétrique discret non local): On se donne  $\mathbf{v} \in V \subset [L^2(\Omega)]^P$ . On appelle  $\mathcal{B}_\theta^h(\mathbf{v})$  un opérateur discret la transformation

$$\mathcal{B}(\mathbf{v})^h(\mathbf{x}) = \frac{1}{n} \sum_{i \in D_h} w_i k_\theta(\mathbf{x}, \mathbf{y}_i, \mathbf{v}(\mathbf{x}), \mathbf{v}(\mathbf{y}_i)) \mathbf{v}(\mathbf{y}_i).$$

**Definition** (discrétisation d'opérateur convergente): On se donne  $\mathbf{v} \in V \subset [L^2(\Omega)]^P$ . On se donne  $\mathcal{B}_\theta(\mathbf{v})$  et un opérateur discret  $\mathcal{B}_\theta^h(\mathbf{v})$ . On dit qu'un d'opérateur discret non local est convergent si

$$\| \mathcal{B}_\theta(\mathbf{x}) - \mathcal{B}_\theta^h(\mathbf{x}) \|_{L_\infty(\Omega)} \xrightarrow[n \rightarrow \infty]{} 0, \quad \forall \mathbf{x} \in \Omega, \forall \mathbf{v} \in V$$

**Definition** (Opérateurs neuronaux): Un opérateur neuronaux sera une composition de discrétisation d'opérateurs paramétriques non locaux et locaux ou l'ensemble des donne une discrétisation convergente d'une composition d'opérateur.

# Représentation spectrale

- Coeur de la méthode: **couche nonlocale**.
- Cas linéaire:

$$u(\mathbf{x}) = \int_{\Omega} k(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y}$$

**Remark** (Strategie spatial):

- une paramétrisation du noyau  $k(\mathbf{x}, \mathbf{y})$  par des fonctions apprenables
- une discrétisation de l'intégrale des méthodes de Monte-Carlo (parfois portée par un graphe) ou des méthodes de quadrature uniformes.

**Definition** (Base Hilbertienne):  $\mathcal{H}$  un espace de Hilbert. Une base hilbertienne est une suite  $(\varphi_j)_{j \geq 1}$  telle que

- $\langle \varphi_i, \varphi_j \rangle = \delta_{ij}$
- pour tout  $f \in \mathcal{H}$  on a

$$f = \sum_{j=1}^{\infty} \langle f, \varphi_j \rangle \varphi_j$$

avec la convergence dans  $\mathcal{H}$ .

## Représentation spectrale II

**Definition** (Spectral approximation of an operator): Soit  $\mathcal{L} : \mathcal{D}(\mathcal{L}) \subset \mathcal{H} \rightarrow \mathcal{H}$  un opérateur linéaire compact auto-adjoint positif sur un espace de Hilbert  $\mathcal{H}$  avec  $\mathcal{D}(\mathcal{L})$  son domaine. Par le théorème spectral on sait qu'il existe une base hilbertienne de vecteurs propres  $(\varphi_j)_{j \geq 1}$  et une suite décroissante de valeurs propres  $(\lambda_j)_{j \geq 1}$  telle que

$$\mathcal{L}\varphi_j = \lambda_j \varphi_j$$

et pour tout  $f \in \mathcal{H}$

$$\mathcal{L}f = \sum_{j=1}^{\infty} \lambda_j \langle f, \varphi_j \rangle \varphi_j$$

avec la convergence dans  $\mathcal{H}$ . Pour  $(\varphi_j)_{j \geq 1}$  on parle d'une base spectrale de  $\mathcal{L}$ .

- **Base de Fourier (conditions périodiques sur  $[0, 1]$ ) :**

$$\varphi_j(x) = \exp(i2\pi jx), \quad j \in \mathbb{Z}$$

Ce sont les vecteurs propres de l'opérateur de Laplace

$$\mathcal{L}y = -y''(x)$$

avec conditions périodiques et avec les valeurs propres  $\lambda_j = (2\pi)^2 j^2$ .

## Representation spectral II

---

- **Polynômes de Legendre (sur  $[-1, 1]$ ) :**

$$\varphi_j(x) = P_j(x)$$

, où  $P_j$  est le polynôme de Legendre de degré  $j$ . Ce sont les vecteurs propres de l'opérateur de Sturm–Liouville

$$\mathcal{L}y = \frac{d[(1-x^2)y'(x)]}{dx}$$

avec valeurs propres  $\lambda_j = j(j+1)$ .

- **Polynômes de Chebyshev (sur  $[-1, 1]$ ) :**  $\varphi_j(x) = T_j(x)$ , où  $T_j$  est le polynôme de Chebyshev de degré  $j$ . Ils satisfont l'équation de Sturm–Liouville

$$(1-x^2)\varphi''(x) - x\varphi'(x) + j^2\varphi(x) = 0,$$

donc sont vecteurs propres de l'opérateur

$$\mathcal{L}y = -(1-x^2)y''(x) + xy'(x),$$

avec valeur propre  $\lambda_j = j^2$ .

# Représentation spectrale III

**Idea:** Faire un changement de base et apprendre dans le domaine spectral.

$$T_k u(\mathbf{x}) = \int_{\Omega} u(\mathbf{x}) \varphi_i(\mathbf{x}) d\mathbf{x} = \int_{\Omega} \left( \int_{\Omega} k(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y} \right) \varphi_i(\mathbf{x}) d\mathbf{x}, \quad i = 1, \dots, r$$

On choisit  $u$  et  $f$  dans notre base spectrale ce qui donne

$$\sum_{j=1}^r \langle u, \varphi_j \rangle \langle \varphi_j, \varphi_i \rangle = \sum_{j=1}^r \langle f, \varphi_j \rangle \left( \int_{\Omega} \left( \int_{\Omega} k(\mathbf{x}, \mathbf{y}) \varphi_j(\mathbf{y}) d\mathbf{y} \right) \varphi_i(\mathbf{x}) d\mathbf{x} \right)$$

ce qui donne en définissant  $u_j = \langle u, \varphi_j \rangle_{L^2}$  et  $f_j = \langle f, \varphi_j \rangle_{L^2}$

$$\mathbf{M}\mathbf{u} = \mathbf{K}\mathbf{f}$$

avec  $M_{ij} = \int_{\Omega} \varphi_j(\mathbf{x}) \varphi_i(\mathbf{x}) d\mathbf{x}$  et  $K_{ij} = \int_{\Omega} \int_{\Omega} k(\mathbf{x}, \mathbf{y}) \varphi_j(\mathbf{y}) \varphi_i(\mathbf{x}) d\mathbf{x} d\mathbf{y}$ .

Les bases spectrales étant orthogonales on a  $\mathbf{M} = \mathbf{I}$  et donc

$$\mathbf{u} = \mathbf{K}\mathbf{f}$$

avec  $\mathbf{u} = (u_1, \dots, u_r)$  et  $\mathbf{f} = (f_1, \dots, f_r)$  les modes de  $u$  et  $f$  dans la base spectrale.



## Spectral neural operator

- Dicitéise par Monte-Carlo  $\rightarrow$  un produit matrice vecteur. Idem en spectral

**Idea:** Si on se donne un opérateur de changement de base  $\mathcal{B}$ , son inverse etalors on a

$$T_k = \mathcal{B}^{-1} T_s \mathcal{B} f$$

avec  $T_s$  le noyau dans le domaine spectrale. L'idée des méthodes spectrales est donc de chercher à paramétrer et apprendre  $T_s$  plutôt que  $T_k$ .

- Deux stratégies: passer totalement en spectral ou faire des allers-retours

**Definition** (Spectral neural operator): Soit  $\mathcal{B}$  un opérateur de changement de base vers une base spectrale tronquée à l'ordre  $r$ . Soit  $f : \Omega \rightarrow \mathbb{R}^{\text{in}}$  une fonction. On appelle un opérateur neuronal spectral

$$\Phi(f) = (\mathcal{B}_r^{-1} \circ L_1 \circ \dots \circ L_1 \circ \mathcal{B}_r) f(\mathbf{x})$$

avec  $\mathcal{B}_r$  la transformée dans la bae limitée à  $r$  modes et  $L_i : \mathbb{R}^{\text{rin}} \rightarrow \mathbb{R}^{\text{rout}}$  réseaux type ML qui vont agir sur l'ensemble des modes.

# FNO I

---

- En pratique l'approche la plus utiliser est
  - de paramétrer l'opérateur non local en spectral
  - de paramettrer le local et la nonlinarité en en space.
- 1er: approche FNO (Fourier Neural Operator)

**Definition** (Continue Fourier layer): Soit  $\mathbf{f}_l : \Omega \rightarrow \mathbb{R}^{\text{in}}$  une fonction avec  $\Omega \in \mathbb{R}^d$ . On se donne un nombre de modes  $r$ . On appelle couche de Fourier l'opération

$$\mathbf{f}_{l+1}(\mathbf{x}) = \sigma((\mathcal{F}_r^{-1}(\mathbf{K}_\theta \cdot \mathcal{F}_r(\mathbf{f}_l)))(\mathbf{x}) + \mathbf{W}\mathbf{f}_l(\mathbf{x}))$$

avec  $\mathcal{F}_r$  les  $r$  premier modes de la transformation de Fourier,  $\mathbf{K}_\theta : \mathbb{C}^{r^d, \text{in}} \rightarrow \mathbb{C}^{r^d, \text{out}}$  un tenseur de coefficient apprenable  $\mathbf{W} : \mathbb{R}^{\text{in}} \rightarrow \mathbb{R}^{\text{out}}$  une couche linéaire locale.

## FNO II

**Definition** (Discrete Fourier layer): Soit  $\mathbf{f}_1 : \Omega \rightarrow \mathbb{R}^{d_v}$  une fonction avec  $\Omega \in \mathbb{R}^d$ . On se donne un nombre de modes  $r$ . On appelle couche de Fourier la suite d'opération évaluée pour un ensemble de point  $\mathbf{x}$  (portée par une grille cartésienne of size  $(s_1, \dots, s_d)$  avec  $n = s_1 \times \dots s_d$ ):

- La FTT

$$\mathbf{a}(\mathbf{k}) = (\mathbf{F}\mathbf{f})(\mathbf{k}) = \sum_{x_1=0}^{s_1-1} \dots \sum_{x_d=0}^{s_d-1} \mathbf{v}_1(x_1, \dots, x_d) \exp\left(-2\pi i \sum_{j=1}^d \frac{x_j k_j}{s_j}\right).$$

- La convolution dans l'espace spectral

$$b_{k,l} = \sum_{j=1}^{d_v} K_{k,l,j} b_{k,j}, k = 1, \dots, k_{\max}, j = 1, \dots, d_v. \quad K \in \mathbb{C}^{r, d_v, d_v}$$

- la FFT inverse

$$\mathbf{c}(\mathbf{x}) = \sum_{k_1=0}^{s_1-1} \dots \sum_{k_d=0}^{s_d-1} \mathbf{b}(k_1, \dots, k_d) \exp\left(2\pi i \sum_{j=1}^d \frac{x_j k_j}{s_j}\right).$$

La somme avec l'opérateur Local

- $\mathbf{v}_{l+1}(\mathbf{x}) = \sigma(\mathbf{c}(\mathbf{x}) + \mathbf{W}\mathbf{v}_l(\mathbf{x}))$

## FNO III

**Proposition** (Fourier layer as an operator neural): La couche discrete converge vers la couche continue et donc est une couche d'opérateur neural car la FFT converge vers la transformée de Fourier et le nombre de modes utilisé est fixe.

**Definition** (Fourier Neural Operator (FNO)): Soit  $\mathbf{f} : \Omega \rightarrow \mathbb{R}^{d_v}$  une fonction avec  $\Omega \in \mathbb{R}^d$ . On appelle FNO:

$$\Phi(\mathbf{f}) = \mathbf{Q} \circ (\mathbf{f}_L \circ \dots \circ \mathbf{f}_1) \circ \mathbf{P}(\mathbf{f})$$

avec  $\mathbf{P} : \mathbb{R}^{\text{in}} \rightarrow \mathbb{R}^{d_v}$  et  $\mathbf{Q} : \mathbb{R}^{d_v} \rightarrow \mathbb{R}^{\text{out}}$  des couches linéaires ou des MLP qui s'applique localement,  $\mathbf{f}_1$  des couches de Fourier et  $L$  le nombre de couche.

Les opérateurs de type FNO l'approche de base pour les opérateurs neuronaux. Cependant il existe un certains nombres de limites:

- Le FNO est défini sur des grilles cartésiennes. Il n'est pas trivial de l'étendre a des grilles non cartésiennes.
- Le FNO est défini sur des domaines périodiques. Il n'est pas trivial de l'étendre a des domaines non périodiques.
- Le FNO est défini sur des domaines euclidiens. Il n'est pas trivial de l'étendre a des domaines courbes.
- Le cout des FNO augmente exponentiellement avec la dimension du domaine.

## FNO IV

Pour les problèmes non périodiques, on utilise deux stratégies:

- une approche de “padding périodique” qui consiste à étendre le domaine de manière périodique.
- une approche de “padding nul” qui consiste à étendre le domaine par des zéros.

**Definition** ( Transformé de Fourier discrete sur un nuage de points): Soit  $\mathbf{f}$  un vecteur d’une fonction discrétisée avec  $n$  points. La transformée de Fourier discrete sur un nuage de points est définie par

$$\mathbf{u} = \mathbf{F}\mathbf{f}$$

avec la matrice  $\mathbf{F} \in \mathbb{C}^{n,r}$  définie par

$$F_{ij} = \frac{1}{d} e^{-2i\pi \langle \mathbf{x}_i, \mathbf{k}_j \rangle}$$

La transformée de Fourier inverse est définie par

$$\mathbf{f} = \mathbf{F}^* \mathbf{u}$$

avec  $\mathbf{F}^*$  la transposée conjuguée de  $\mathbf{F}$ .

Cela marche mais on perd l’avantage de la FFT rapide.

## Factorized FNO

- Nombre de poids en dimension  $d$ :  $r^d \times d_v \times d_u$ .
- Simplifications. Prenons notre opérateur de Laplace  $-\Delta u = f$  en 2D et supposons que la solution est de rang faible

$$u(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^r g_i(\mathbf{x}_1) h_i(\mathbf{x}_2)$$

- Si on applique l'opérateur de Laplace on a

$$f(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^r (-g_i''(\mathbf{x}_1) h_i(\mathbf{x}_2) - g_i(\mathbf{x}_1) h_i''(\mathbf{x}_2))$$

qui est aussi de rang faible  $r$ . Il s'agit donc d'un opérateur qui **conserve la structure de rang faible**. On se donne une fonction separable  $f(\mathbf{x}, \mathbf{y}) = g(\mathbf{x})h(\mathbf{y})$ . Si on applique une transformation de fonction 2D cela s'écrit

$$\begin{aligned} \mathcal{F}(f)(\mathbf{k}_x, \mathbf{k}_y) &= \int \int f(\mathbf{x}, \mathbf{y}) e^{-2i\pi(\mathbf{x}\mathbf{k}_x + \mathbf{y}\mathbf{k}_y)} d\mathbf{x} d\mathbf{y} \\ &= \left( \int g(\mathbf{x}) e^{-2i\pi\mathbf{x}\mathbf{k}_x} d\mathbf{x} \right) \left( \int h(\mathbf{y}) e^{-2i\pi\mathbf{y}\mathbf{k}_y} d\mathbf{y} \right) \\ &= \mathcal{F}(g)(\mathbf{k}_x) \mathcal{F}(h)(\mathbf{k}_y) \end{aligned}$$

Donc la transformé de Fourier d'une fonction séparable est aussi séparable et idem pour la fonction separable.

## Factorized FNO II

---

- Par linéarité la **transformée de fourier préserve le rang**.
- A partir du moment on construit des filtres separable on voit bien que notre convolution spectrale va conserver la structure de rang faible.

**Definition** (Couche continue factorized FNO): Soit  $\mathbf{f}_l : \Omega \rightarrow \mathbb{R}^{\text{in}}$  une fonction avec  $\Omega \in \mathbb{R}^d$ . On se donne un nombre de modes  $r$ . On appelle couche factorized FNO l'opération

$$\mathbf{f}_{l+1}(\mathbf{x}) = \sigma \left( \sum_{i=1}^d \left( \mathcal{F}_{\mathbf{x}_i, r}^{-1} \left( \mathbf{K}_{\boldsymbol{\theta}_i} \cdot \mathcal{F}_{\mathbf{x}_i, r}(\mathbf{f}_l) \right) \right) (\mathbf{x}) + \mathbf{W} \mathbf{f}_l(\mathbf{x}) \right)$$

- Cette couche n'applique que des convolutions 1D dans chaque direction.
- Nombre de paramètre:  $n_{\text{param}} = d * (r * d_{\text{in}} * d_{\text{out}})$  ce qui la rend de  $d$  fois moins coûteuse qu'une couche FNO classique.

# FNO extension to general geometry I

- La première approche est de se ramener a un FNO sur des domaines euclidiens.
- Pour cela:
  - Faire le changement de coordonnées du domaine complexe au domaine euclidien
  - Appliquer un FNO dans le domaine euclidien
  - Revenir dans le domaine complexe

**Definition** (Transformée de Fourier avec mapping): On se donne une fonction inversible  $\varphi$  qui part du domaine de calcul  $= [a, b]^d$  vers le domaine physique  $\Omega$  Soit une fonction de  $f : \Omega \rightarrow \mathbb{R}^p$  on définit la transformée de Fourier avec mapping par

$$\begin{aligned}\hat{\mathbf{f}}(\mathbf{k}) &= \mathcal{F}_{\varphi} \mathbf{f}(\mathbf{x}) = \int_{\mathbf{D}} \mathbf{f}(\mathbf{x}) e^{-2i\pi \langle \mathbf{x}_c, \mathbf{k} \rangle} d\mathbf{x} \\ &= \int_{\Omega} \mathbf{f}(\varphi(\mathbf{y})) e^{-2i\pi \langle \varphi^{-1}(\mathbf{y}), \mathbf{k} \rangle} |\det(\nabla_{\mathbf{x}} \varphi^{-1})(\mathbf{y}))| d\mathbf{y}\end{aligned}$$

et la transformée de Fourier inverse est donnée par

$$\mathbf{f}(\mathbf{x}) = \mathcal{F}_{\varphi}^{-1} \hat{\mathbf{f}}(\mathbf{k}) = \int_{\mathbf{D}} \hat{\mathbf{f}}(\mathbf{k}) e^{2i\pi \langle \varphi^{-1}(\mathbf{x}), \mathbf{k} \rangle} d\mathbf{k}$$



# FNO extension to general geometry II

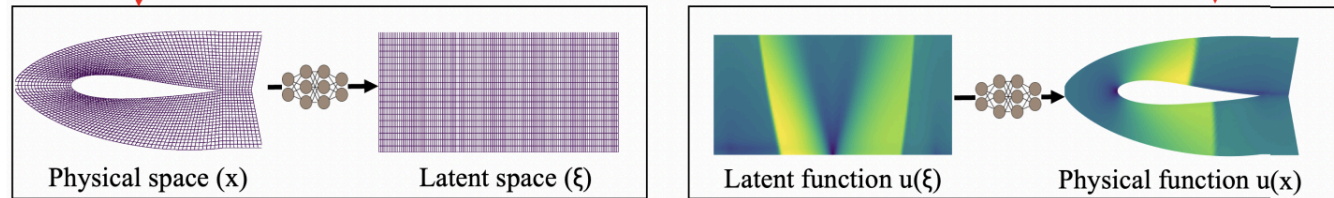
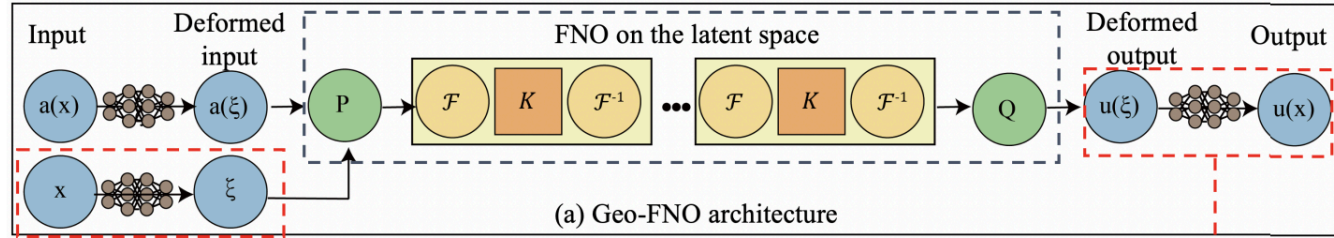
**Definition** (Architecture Geo-FNO): Soit  $\mathbf{f} : \Omega \rightarrow \mathbb{R}^{d_v}$  une fonction avec  $\Omega \in \mathbb{R}^d$ . On appelle Geo-FNO:

$$\Phi(\mathbf{f}) = \mathbf{Q} \circ (\mathbf{f}_L \circ \dots \circ \mathbf{f}_1) \circ \mathbf{P}(\mathbf{f})$$

avec  $\mathbf{P} : \mathbb{R}^{\text{in}} \rightarrow \mathbb{R}^{d_v}$  et  $\mathbf{Q} : \mathbb{R}^{d_v} \rightarrow \mathbb{R}^{\text{out}}$  des couches linéaires ou des MLP qui s'applique localement,  $\mathbf{f}_i$  des couches de Fourier. Ici

$$\mathbf{f}_i = \sigma\left(\left(\mathcal{F}_r^{-1}(K_\theta \cdot \mathcal{F}_r(\mathbf{f}_i))\right)(\mathbf{x}) + W\mathbf{f}_i(\mathbf{x})\right), \quad \forall i = 2, \dots, L-1$$

$$\mathbf{f}_1 = \sigma\left(\left(\mathcal{F}_r^{-1}(K_\theta \cdot \mathcal{F}_{\varphi,r}(\mathbf{f}_1))\right)(\mathbf{x}) + W\mathbf{f}_1(\mathbf{x})\right), \quad \mathbf{f}_L = \sigma\left(\left(\mathcal{F}_{r,\varphi}^{-1}(K_\theta \cdot \mathcal{F}_r(\mathbf{f}_1))\right)(\mathbf{x}) + W\mathbf{f}_1(\mathbf{x})\right)$$



(b) Deformation

## FNO extension to general geometry III

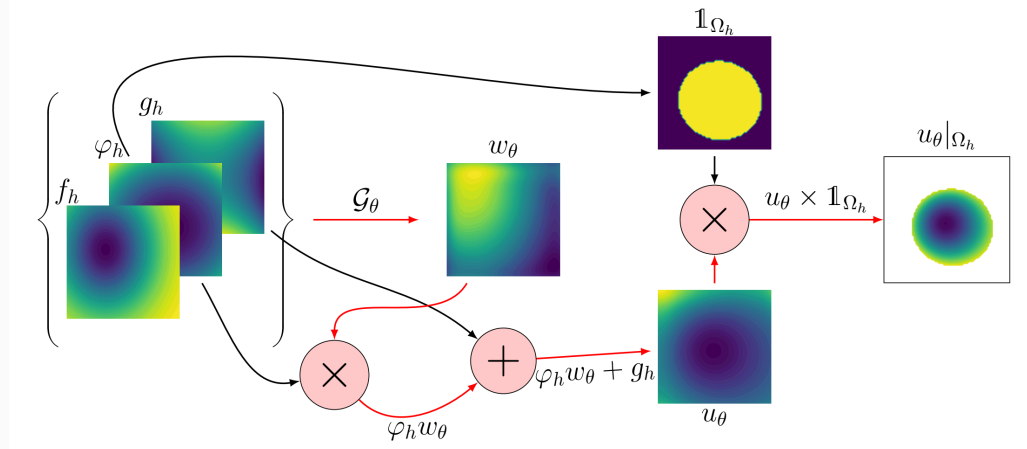
- Ici la geometrie complexe est prise en compte par le mapping  $\varphi$ .
- On peut aussi utiliser une approche de type fonction de niveau comme pour les PINNs
- La méthode  $\varphi$ -FNO en pratique ne change pas l'architecture standard du FNO mais donne en entrée la géométrie sous forme de fonction.
- On se donne une level set du domaine  $\varphi$  une source  $\mathbf{f}$  et une source de bord  $\mathbf{g}$ .

**Definition** (Architecture  $\varphi$ -FNO): Soit  $\mathbf{f} : \Omega \rightarrow \mathbb{R}^{d_v}$  une fonction avec  $\Omega \in \mathbb{R}^d$ . On appelle  $\varphi$ -FNO l'opération

$$\Phi(\mathbf{f}) = \Phi_{\text{po}} \circ Q \circ (\mathbf{f}_L \circ \dots \circ \mathbf{f}_1) \circ P(\mathbf{f}, \mathbf{g}, \varphi)$$

avec  $P : \mathbb{R}^{\text{in}} \rightarrow \mathbb{R}^{d_v}$  et  $Q : \mathbb{R}^{d_v} \rightarrow \mathbb{R}^{\text{out}}$  des couches linéaires ou des MLP qui s'applique localement,  $\mathbf{f}_1$  des couches de Fourier,  $\Phi_{\text{po}}$  est donnée par:

$$\Phi_{\text{po}}(\mathbf{w}_\theta(\mathbf{x})) = (\mathbf{w}_\theta(\mathbf{x})\varphi(\mathbf{x}) + \mathbf{g}(\mathbf{x}))\chi_{\varphi(\mathbf{x}) \leq 0}$$



## FNO extension to general geometry IV

- Dernière idée: utiliser des moyennes locales pour passer du domaine physique à un domaine carré de référence qui contient le domaine physique.

**Definition** (Local averaging operator): Soit  $\mathcal{M} \in \mathbb{R}^d$  une variété de  $\mathbb{R}^d$  et  $\Omega \in \mathbb{R}^d$  un sous espace . Soit  $\mathbf{f} : \mathcal{M} \rightarrow \mathbb{R}^{d_v}$  une fonction. On définit l'opérateur de moyenne locale de l'espace physique vers l'espace latent par

$$\mathcal{E}_\varepsilon(\mathbf{f})(\mathbf{x}) = \frac{1}{|\mathbf{B}(\mathbf{x}, \varepsilon)|} \int_{\mathbf{B}(\mathbf{x}, \varepsilon) \cap \mathcal{M}} \mathbf{f}(\mathbf{y}) d\mathbf{y}, \quad \forall \mathbf{x} \in \Omega$$

avec  $\mathbf{B}(\mathbf{x}, \varepsilon)$  la boule de centre  $\mathbf{x}$  et de rayon  $\varepsilon$  et l'opérateur de moyenne locale de l'espace latent vers l'espace physique par

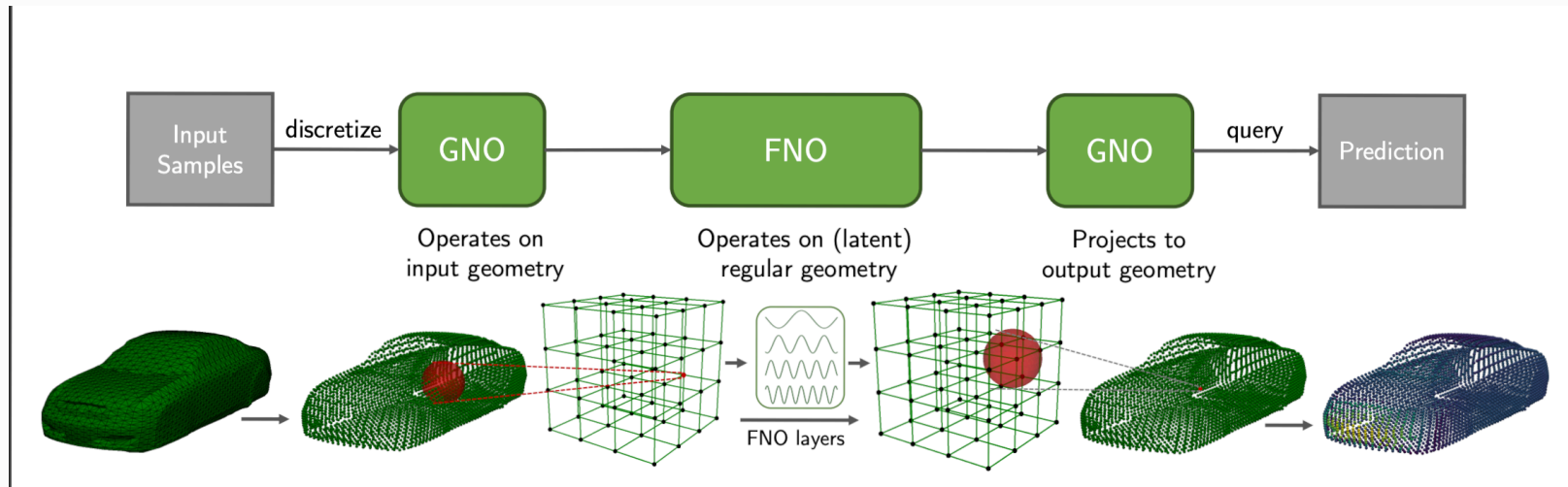
$$\mathcal{D}_\varepsilon(\mathbf{f})(\mathbf{x}) = \frac{1}{|\mathbf{B}(\mathbf{x}, \varepsilon)|} \int_{\mathbf{B}(\mathbf{x}, \varepsilon)} \mathbf{f}(\mathbf{y}) d\mathbf{y}, \quad \forall \mathbf{x} \in \mathcal{M}$$

# FNO extension to general geometry V

**Definition** (Architecture GINO): Soit  $f : \Omega \rightarrow \mathbb{R}^{d_v}$  une fonction avec  $\Omega \in \mathbb{R}^d$ . On appelle GINO l'opération

$$\Phi(f) = \mathcal{D}_\varepsilon \circ \text{FNO} \circ \mathcal{E}_\varepsilon(f)$$

avec  $P : \mathbb{R}^{\text{in}} \rightarrow \mathbb{R}^{d_v}$  et  $Q : \mathbb{R}^{d_v} \rightarrow \mathbb{R}^{\text{out}}$  des couches linéaires ou des MLP qui s'applique localement,  $f_l$  des couches de Fourier et  $\varepsilon$  un hyper-paramètre.



## 2. Operator learning: Manifold spectral approach

# Spectral basis in general domain and manifold

---

Pour des géométries plus générales on a deux cas:

- Domaine général dans  $\mathbb{R}^d$
- Variété dans  $\mathbb{R}^d$
- Les méthodes qu'on a introduit précédemment étaient plutôt adaptées aux domaines généraux mais sans courbure (seul le bord est courbe).
- 

## Remark:

- Geo-FNO utilise un mapping globale du domaine complexe vers un domaine euclidien simple. Ce mapping globale n'existe pas pour beaucoup de variété
- GINO utilise une moyenne locale dans le domaine physique pour se ramener à un domaine euclidien. Cependant elle va passer d'une variété 2D à un domaine 3D ce qui n'est pas optimal.

- On va proposer une approche spectrale directe pour les variétés.
- Cette approche pourrait être aussi utilisée pour des domaines généraux mais avec des adaptations théoriques.

## Spectral basis in general domain and manifold

**Remark:** La deuxième approche consiste à appliquer une méthode spectrale directement sur le domaine complexe ou la variété. Quelle base ?

- Caractéristique de la base de Fourier: Base de fonction de plus en plus **oscillante**. On part de la fonction constante puis une fonction qui oscille une fois, deux fois, etc.

**Remark:** La restriction des modes classiques à  $\Omega$  ne donne pas forcément des fonctions **orthogonales** de plus en plus oscillantes sur  $\Omega$ . “La base de Fourier dépend de la géométrie du domaine (conditions aux limites, forme du domaine, métrique...)”



**Figure 3:** Some functions of the Manifold Harmonic Basis (MHB). Note the similarity with the sine products used by the DCT.

# Riemannian geometry

---

**Definition** (Variété Riemannienne): Soit  $\mathcal{M}$  une variété différentielle de dimension  $d$ . On dit que  $\mathcal{M}$  est une variété riemannienne si elle est munie d'un champ de tenseurs  $g$ , appelé métrique riemannienne, qui associe à chaque point  $\mathbf{x} \in \mathcal{M}$  un produit scalaire  $g_{\mathbf{x}} : T_{\mathbf{x}}\mathcal{M} \times T_{\mathbf{x}}\mathcal{M} \rightarrow \mathbb{R}$  défini positif.

La condition de régularité est que, pour tout couple de champs de vecteurs lisses  $X, Y$  sur  $\mathcal{M}$ , la fonction  $\mathbf{x} \rightarrow g_{\mathbf{x}}(X(\mathbf{x}), Y(\mathbf{x}))$  est une fonction  $C^\infty$  sur  $\mathcal{M}$ .

Ce produit scalaire permet de mesurer :

- la **longueur** d'un vecteur tangent  $\mathbf{v}$  par

$$\|\mathbf{v}\|_g = \sqrt{g_{\mathbf{x}}(\mathbf{v}, \mathbf{v})},$$

- l'**angle** entre deux vecteurs tangents  $\mathbf{v}$  et  $\mathbf{w}$  par

$$\cos(\theta) = \frac{g_{\mathbf{x}}(\mathbf{v}, \mathbf{w})}{\|\mathbf{v}\|_g \|\mathbf{w}\|_g},$$



## Riemanian geometry II

---

Maintenant considérons une variété paramétrée. Donc on a

$$\mathcal{M} = \mathbf{x} = \varphi(\mathbf{u}), \quad \mathbf{u} \in U \subset \mathbb{R}^d, \}$$

$$T_{\varphi(\mathbf{u})}\mathcal{M} = \text{span}\{\partial_{u_1}\varphi(\mathbf{u}), \dots, \partial_{u_d}\varphi(\mathbf{u})\}.$$

La métrique riemannienne induite sur  $\mathcal{M}$  par la métrique euclidienne de  $\mathbb{R}^n$  s'écrit, en coordonnées locales  $\mathbf{u}$ , comme la matrice symétrique définie positive :

$$g_{ij}(\mathbf{u}) = \partial_{u_i}\varphi(\mathbf{u}) \cdot \partial_{u_j}\varphi(\mathbf{u}), \quad g(\mathbf{u}) = (D\varphi(\mathbf{u}))^T D\varphi(\mathbf{u}).$$

Il s'agit de la matrice du produit de chaque fonction de base de  $T\mathcal{M}$  avec les autres.

- Considérons  $\varphi : (\theta, \varphi) \mapsto (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)$  une paramétrisation de  $S^2 \subset \mathbb{R}^3$ .

On a :

$$\partial_{\theta}\varphi = (\cos \theta \cos \varphi, \cos \theta \sin \varphi, -\sin \theta), \quad \partial_{\varphi}\varphi = (-\sin \theta \sin \varphi, \sin \theta \cos \varphi, 0).$$

$$g = \begin{pmatrix} g_{\theta\theta} & g_{\theta\varphi} \\ g_{\varphi\theta} & g_{\varphi\varphi} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \sin^2 \theta \end{pmatrix}.$$

## Riemannian geometry III

- On détaille pas la construction mais on peut définir un Laplacien sur la variété

**Definition** (Laplacien de Laplace-Beltrami): Le **laplacien de Laplace-Beltrami** est l'opérateur différentiel d'ordre deux défini par en coordonnées locales  $(x^1, \dots, x^d)$ , on obtient :

$$\Delta f = \frac{1}{\sqrt{|g|}} \partial_i \left( \sqrt{|g|} g^{ij} \partial_j f \right).$$

**Definition** (Base spectrale d'une variété riemannienne): Soit  $\mathcal{M}$  une variété riemannienne compacte sans bord. On appelle base spectrale de  $\mathcal{M}$  l'ensemble des fonctions propres du laplacien de Laplace-Beltrami

$$-\Delta \varphi_{\mathcal{M},j} = \lambda_j \varphi_{\mathcal{M},j}, \quad j = 1, 2, \dots$$

avec  $0 = \lambda_1 < \lambda_2 < \dots$  le spectre du laplacien de Laplace-Beltrami.

## Riemannian geometry IV

**Definition** (Opérateur spectral à noyau sur une variété riemannienne): Soit  $\mathcal{M}$  une variété riemannienne compacte sans bord. Soit  $k : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$  une fonction lisse. On définit l'opérateur à noyau dans le domaine spectral par

$$\hat{u}(k) = \sum_{j=1}^r K_j \hat{f}(j)$$

avec  $\hat{f}(j) = \langle f, \varphi_{\mathcal{M},j} \rangle = \int_{\mathcal{M}} f(x) \varphi_{\mathcal{M},j}(x) dx$  et  $\hat{u}(j) = \langle u, \varphi_{\mathcal{M},j} \rangle = \int_{\mathcal{M}} u(x) \varphi_{\mathcal{M},j}(x) dx$ .

- On voit donc que si on connaît les vecteurs propres du laplacien de Laplace-Beltrami, on peut exprimer l'opérateur à noyau ou les convolutions en termes de ces vecteurs propres.

**Remark:** Comme pour le cas Euclidien les modes forment une base orthogonale et de plus en plus oscillante sur la variété lorsque l'on augmente les valeurs propres (qui sont les fréquences).

# Spectral graph geometry and Manifold I

Dans la suite on va:

- introduire des approximations du laplacien de Laplace-Beltrami sur des graphes
- introduire des architectures de GNN/GNO spectraux et discuter de la convergence.

**Definition** (Dérivée): L'opérateur dérivée  $d : \mathcal{H}(V, \chi) \rightarrow \mathcal{H}(E, \psi)$  peut être défini par

$$\forall e_{ij} \in E, \quad d(e_{ij}) = \gamma(w_{ij})(f(j) - f(i))$$

avec  $f(i), f(j)$  la valeur de la fonction  $f$  aux sommets  $i$  et  $j$ . On peut aussi le définir avec:

$$\forall e_{ij} \in E, \quad d^n(e_{ij}) = \gamma(w_{ij}) \left( \frac{f(j)}{\sqrt{d(j)}} - \frac{f(i)}{\sqrt{d(i)}} \right)$$

avec  $d(\cdot)$  le degré du noeud et  $\gamma : \mathbb{R}_+^* \rightarrow \mathbb{R}_+^*$ .

**Definition** (Dual de la dérivée): Le dual de l'opérateur dérivé  $d$  est donné par

$$\forall v_j \in V, \quad d^*(v_j) = \frac{1}{2n\chi(d_j)} \sum_{i=1} \gamma(w_{ij})\psi(w_{ij})(g_{ij} - g_{ji})$$

avec  $\psi(x) > 0$  sur  $\mathbb{R}_+^*$  et  $\psi(0) = 0$ . Par analogie avec le calcul différentiel classique on peut le voir comme un opérateur de divergence.

# Spectral graph geometry and Manifold II

**Definition** (Laplacien sur un graphe): L'opérateur **laplacien sur un graphe**  $\Delta : \mathcal{H}(V, \chi) \rightarrow \mathcal{H}(V, \psi)$  est défini par  $\Delta(i) = d^*d$  On peut obtenir une formule explicite:

$$\Delta(i) = \frac{1}{\chi(i)} \left[ f(i) \frac{1}{n} \sum_{j=1}^n \gamma(w_{ji})^2 \psi(w_{ji}) - \frac{1}{n} \sum_{j=1}^n f(j) \gamma(w_{ji})^2 \psi(w_{ji}) \right]$$

On peut utiliser aussi  $d^n$  et son dual.

- 3 Laplaciens sur graphe classique dans la littérature:

- Le Laplacien dit **Marche aléatoire**:

$$\Delta^{rw}f = (I_d - D^{-1}W)f$$

obtenu avec  $\frac{\gamma(w_{ji})^2 \psi(w_{ji})}{\chi(d_i)} = \frac{w_{ij}}{d_i}$ .

- Le Laplacien dit **non normalisé**:

$$\Delta^u f = (D - W)f$$

obtenu avec  $\frac{\gamma(w_{ji})^2 \psi(w_{ji})}{\chi(d_i)} = w_{ij}$ .

- Le Laplacien dit **normalisé**:

$$\Delta^n f = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}f$$

obtenu avec  $d^n$  au lieu de  $d$  et  $\gamma(w_{ji})^2 \psi(w_{ji}) = w_{ij}$  et  $\chi(d_i) = 1$ .

## Spectral graph geometry and Manifold III

**Definition** (Opérateur de Laplace-Beltrami à poids): Soit  $\mathcal{M}$  une variété riemannienne compacte sans bord. Soit  $p(\mathbf{x})$  une densité de probabilité sur la variété par rapport aux volumes naturels  $dV = \sqrt{\det g} dx$ . On définit l'opérateur de Laplace-Beltrami à poids par Soit  $s \in \mathbb{R}$  le Laplacien à poids est défini par  $\Delta_s f = \frac{1}{p^s} \nabla \cdot (p^s \nabla f)$ .

**Theorem** (Convergence des laplaciens sur graphes): Soit  $\mathcal{M}$  une variété riemannienne de dimension  $m$  incluse dans  $\mathbb{R}^d$ . Soit  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)$  un échantillon obtenu avec une loi de probabilité  $P$  et de densité positive  $p(\mathbf{x})$  de support  $\mathcal{M}$ . Soit  $\mathbf{x}$  à l'intérieur de la variété. On se donne un graphe  $G = [X, W]$  obtenu par la méthode des  $K$  plus proches voisins tel que si  $\|\mathbf{x}_i - \mathbf{x}_j\| < C(k)h$  alors l'arrête existe dans le graphe. Sous certaines hypothèses sur  $\mathcal{M}$ , l'algorithme des plus proches voisins et sur  $p$ , on a quel si  $h \rightarrow 0$  et  $\frac{nh^{(m+2)}}{\log n} \rightarrow \infty$  alors

- $\Delta^{rw} f$  converge presque sûrement vers  $-C \Delta_s f$
- $\Delta^u f$  converge presque sûrement vers  $-C p^{(1-2\lambda)} \Delta_s f$  avec  $s = (1 - 2\lambda)$  et si  $\frac{nh^{m+4}}{\log n} \rightarrow \infty$  on a
- $\Delta^n f$  converge presque sûrement vers  $-C p^{\frac{1}{2}-\lambda} \Delta_s \frac{f}{p^{\frac{1}{2}-\lambda}}$  avec  $C$  une constante dépendante de  $\lambda$  et des paramètres de l'algorithme de voisin.

# Spectral graph geometry and Manifold IV

- Comment converger vers un Laplacien independant de la densité ?

**Definition** (Noyau integral et operateur de Markov): Soit  $\mathcal{M}$  une variété riemannienne compacte sans bord de dimension  $m$ . Soit  $k_\varepsilon : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$  un noyau symétrique positif défini par

$$k_\varepsilon(\mathbf{x}, \mathbf{y}) = e^{-\left(\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\varepsilon}\right)}$$

avec  $h : \mathbb{R}_+^* \rightarrow \mathbb{R}_+^*$  une fonction décroissante rapide. On définit l'opérateur intégral par

$$\mathcal{K}_\varepsilon f(\mathbf{x}) = \int_{\mathcal{M}} k_\varepsilon(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) dV(\mathbf{y})$$

Soit un densité de probabilité  $p(\mathbf{x})$  sur  $\mathcal{M}$ . On définit l'approximation regulière de  $p$  par  $p_\varepsilon(\mathbf{x}) = \mathcal{K}_\varepsilon p(\mathbf{x})$ . On définir le noyau normalisé par

$$k_\varepsilon^\alpha(\mathbf{x}, \mathbf{y}) = \frac{k_\varepsilon(\mathbf{x}, \mathbf{y})}{p_\varepsilon^\alpha(\mathbf{x}) p_\varepsilon^\alpha(\mathbf{y})}$$

avec  $\alpha \in [0, 1]$ . On définit l'opérateur de Markov par

$$P_\varepsilon^\alpha f(\mathbf{x}) = \int_{\mathcal{M}} \frac{k_\varepsilon^\alpha(\mathbf{x}, \mathbf{y})}{d_\varepsilon^\alpha(\mathbf{x})} f(\mathbf{y})$$

avec  $d_\varepsilon^\alpha(\mathbf{x}) = \int_{\mathcal{M}} k_\varepsilon^\alpha(\mathbf{x}, \mathbf{y}) p(\mathbf{y}) dV(\mathbf{y})$ .

# Spectral graph geometry and Manifold V

---

**Theorem** (Convergence de l'opérateur de Markov): Soit  $\mathcal{M}$  une variété riemannienne compacte sans bord de dimension  $m$ . Soit  $p(\mathbf{x})$  une densité de probabilité sur  $\mathcal{M}$ . Soit  $P_\varepsilon^\alpha$  l'opérateur de Markov défini précédemment. On a

$$\lim_{\varepsilon \rightarrow \infty} \frac{f(\mathbf{x}) - P_\varepsilon^\alpha f(\mathbf{x})}{\varepsilon} = \frac{\Delta(f(\mathbf{x})q^{1-\alpha}(\mathbf{x}))}{q^{1-\alpha}(\mathbf{x})} - \frac{\Delta(q^{1-\alpha}(\mathbf{x}))}{q^{1-\alpha}(\mathbf{x})}f(\mathbf{x})$$

Avec  $\alpha = 1$  on obtient la convergence vers le Laplacien de Laplace-Beltrami classique.



## Spectral graph geometry and Manifold V

**Algorithm** (Construction du laplacien sur graphe convergent vers Laplace-Beltrami): Soit  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)$  un échantillon obtenu avec une loi de probabilité  $P$  et de densité positive  $p(\mathbf{x})$  de support  $\mathcal{M}$ .

- Soit  $W \in \mathbb{R}^{n \times n}$  la matrice des poids définie par

$$W_{ij} = e^{-\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\varepsilon}\right)}$$

- On construit ensuite le noyau normalisé par

$$W^\alpha = Q^{-\alpha} W Q^{-\alpha}, \quad \text{avec} \quad Q_{ii} = \sum_{j=1}^n W_{ij}$$

- On construit la matrice de degré normalisée par

$$D_{ii}^\alpha = \sum_{j=1}^n W_{ij}^\alpha$$

- On construit l'opérateur de Markov discret par

$$P^1 = D^{\alpha, -1} W^\alpha$$

- On construit le laplacien de Laplace-Beltrami discret par

$$L = \frac{I_d - P^1}{\varepsilon}$$

Maintenant on diffèrent graphe Laplacien dont un convergent on peut construire des couches spectrales.

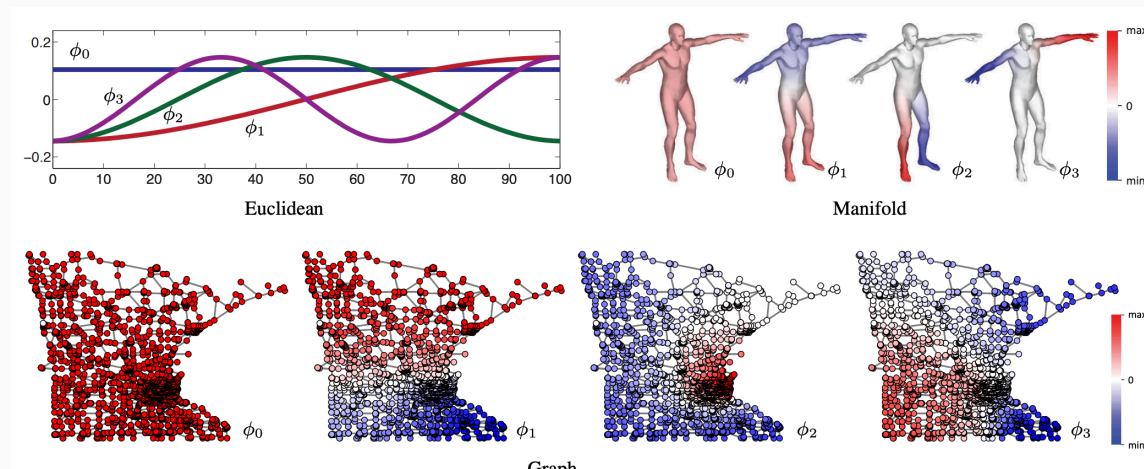
# Spectral GNN I

- On peut démontrer que les matrices  $L$  du graphe Laplacien sont symétrique et définie positive. On peut donc les diagonaliser et obtenir des vecteurs propres orthogonaux.

**Definition** (Opérateur à noyau spectral sur graphe): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $L$  le laplacien sur graphe. Soit  $(\Phi, \Lambda)$  les vecteurs propres et valeurs propres de  $L$ . Soit  $f \in \mathcal{H}(V, \chi)$  un signal sur le graphe. On définit un opérateur à noyau spectral par

$$u = \Phi K \Phi^T f$$

avec  $K$  une matrice. Si on prend  $K$  diagonale on obtient une convolution spectrale.



## Spectral GNN II

**Definition** (Couche de type FNO sur graphe): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $L$  le laplacien sur graphe. Soit  $(\Phi, \Lambda)$  les vecteurs propres et valeurs propres de  $L$ . Soit  $f : V \rightarrow \mathbb{R}^{d_{\text{in}}}$  une fonction sur le graphe et  $g : V \rightarrow \mathbb{R}^{d_{\text{out}}}$  une fonction sur le graphe. On définit une couche de type FNO par

$$g_j = \sigma \left( \sum_{i=1}^q \Phi_r K_{\theta, ij} \Phi_r^T f_i + Wf \right)$$

avec  $K_{\theta, i, j} \in \mathcal{M}_{r, r}(\mathbb{R})$  une matrice apprise,  $\Phi_r$  les  $r$  premiers modes de  $L$  et  $W$  une couche linéaire locale. .

**Proposition:** Si le graphe Laplacien converge vers le Laplacien de Laplace-Beltrami sur une variété riemannienne compacte sans bord alors la couche de type FNO sur graphe converge vers une couche spectrale sur la variété riemannienne. On a alors un opérateur neuronale.

- Cette architecture est totalement naturelle pour des EDP elliptiques car les effets sont globaux.
- EDP hyperboliques ou parabolique l'information locale est très importante.
- Autres applications sur les graphes on souhaite des filtres locaux comme dans les CNN.

# Spectral GNN III

**Remark** (Dualité régularité-localité): Il y a une dualité entre la régularité dans le domaine spectral et la localisation dans le domaine spatial et vice-versa.

**Definition** (Polynomial spectral filters sur graphe): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $L$  le laplacien sur graphe. Soit  $(\Phi, \Lambda)$  les vecteurs propres et valeurs propres de  $L$ . On définit un filtre polynomial par

$$K_{\theta,ij} = \text{diag}(g_{\theta,ij}(\lambda_1), \dots, g_{\theta,ij}(\lambda_r))$$

avec

$$g_{\theta,ij}(\lambda) = \sum_{k=1}^r \theta_k^{ij} \lambda^k$$

et  $\theta_{ij} \in \mathbb{R}^r$  les paramètres appris.

**Lemma** (Filtre sur le Laplacien): Soit un filtre polynomial  $K_{\theta,ij}$  défini précédemment. On a

$$u = \Phi K_{\theta,ij} \Phi^T f = \sum_{k=0}^r \theta_k^{ij} L^k f$$

## Spectral GNN IV

**Remark** (Localité des filtres): On remarque que  $L^k$  est un opérateur qui ne fait interagir que les noeuds a une distance au plus  $k$  dans le graphe. Par conséquent les filtres polynomiaux sont des filtres locaux dont le rayon d'action dépend du degré du polynome.

**Definition** (Graph shift operator): Soit  $G = [V, E]$  un graphe avec  $|V| = n$ . On appelle opérateur de décalage sur graphe (graph shift operator) une matrice  $S \in \mathbb{R}^{n \times n}$  telle que  $S_{ij} \neq 0$  si et seulement si  $(i, j) \in E$  ou  $i = j$ .

**Definition** (Graph signal shift): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $S \in \mathbb{R}^{n \times n}$  un opérateur de décalage sur graphe. Soit  $\mathbf{f} \in \mathcal{H}(V, \chi)$  un signal sur le graphe. On définit le décalage du signal par

$$S\mathbf{f}$$

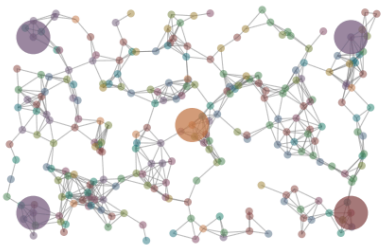
avec  $(S\mathbf{f})(i) = \sum_{j=1}^n S_{ij}\mathbf{f}(j)$ . Typiquement  $S = W$  la matrice de poids ou  $S = L$  le laplacien sur graphe.

# Spectral GNN V

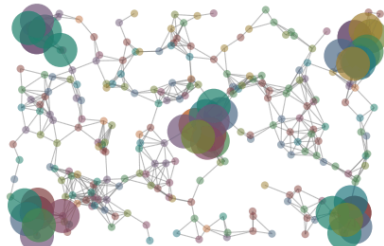
**Definition** (Graph convolutional filter): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $S \in \mathbb{R}^{n \times n}$  un opérateur de décalage sur graphe. Soit  $\mathbf{f} \in \mathcal{H}(V, \chi)$  un signal sur le graphe. On définit un filtre convolutionnel sur graphe par

$$H(S)\mathbf{f} = \sum_{k=0}^K \theta_k S^k \mathbf{f}$$

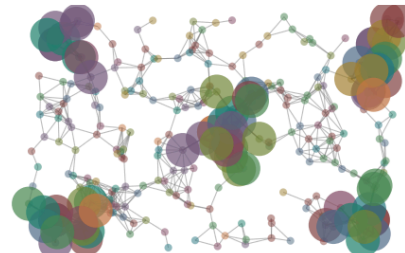
avec  $\theta \in \mathbb{R}^K$  les paramètres appris et  $H(S) = \sum_{k=0}^K \theta_k S^k \in \mathbb{R}^{n \times n}$ .



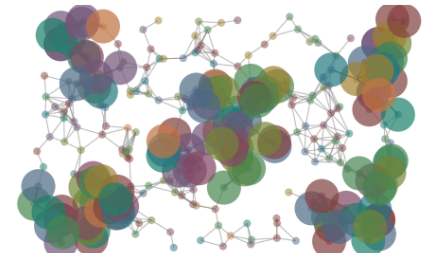
$$\mathbf{x}^{(0)} = \mathbf{x} = \mathbf{S}^0 \mathbf{x}$$



$$\mathbf{x}^{(1)} = \mathbf{S}\mathbf{x}^{(0)} = \mathbf{S}^1 \mathbf{x}$$



$$\mathbf{x}^{(2)} = \mathbf{S}\mathbf{x}^{(1)} = \mathbf{S}^2 \mathbf{x}$$



$$\mathbf{x}^{(3)} = \mathbf{S}\mathbf{x}^{(2)} = \mathbf{S}^3 \mathbf{x}$$

## Spectral GNN VI

**Proposition** (Invariance par permutation des noeuds): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $S \in \mathbb{R}^{n \times n}$  un opérateur de décalage sur graphe. Soit  $H(S)$  un filtre convolutionnel sur graphe. Soit  $P \in \mathbb{R}^{n \times n}$  une matrice de permutation qui appartient à l'espace

$$\left\{ P \in \mathbb{R}^{n \times n} \mid \forall i, j \in \{1, \dots, n\}, P_{ij} \in \{0, 1\}, \sum_{i=1}^n P_{ij} = 1, \sum_{j=1}^n P_{ij} = 1 \right\}$$

On a

$$H(PSP^T)(Pf) = P(H(S)f)$$

- On a

$$H(PSP^T)(Pf) = \left( \sum_{k=0}^K \theta_k (PSP^T)^k \right) (Pf)$$

- Pour obtenir cela on utilise le fait que  $PP^T = I_d$  et donc que

$$(PSP^T)^k = PS^kP^T, \quad \text{donc} \quad H(PSP^T)(Pf) = \left( \sum_{k=0}^K P\theta_k S^k \right) P^T Pf = P \left( \sum_{k=0}^K \theta_k S^k \right) f$$

ce qui donne  $H(PSP^T)(Pf) = PH(S)f$ .

## Spectral GNN VI

**Definition** (Spectral graph convolutional layer): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $S \in \mathbb{R}^{n \times n}$  un opérateur de décalage sur graphe. Soit  $(\Phi, \Lambda)$  les vecteurs propres et valeurs propres de  $L$ . Soit  $\mathbf{f} : V \rightarrow \mathbb{R}^{d_{\text{in}}}$  une fonction sur le graphe et  $\mathbf{g} : V \rightarrow \mathbb{R}^{d_{\text{out}}}$  une fonction sur le graphe. On définit une couche spectrale par

$$\mathbf{g}_j = \sigma \left( \sum_{i=1}^{d_{\text{in}}} H_{\theta, ij}(S) \mathbf{f}_i \right), \quad \forall j \in \{1, \dots, d_{\text{out}}\}$$

avec  $H_{\theta, i, j}(S)$  des filtres convolutionnels sur graphe,  $\sigma$  une activation nonlinéaire.

**Proposition** (Invariance par permutation des noeuds des couches spectrales sur graphe): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $S \in \mathbb{R}^{n \times n}$  un opérateur de décalage sur graphe. Soit  $\mathbf{f} : V \rightarrow \mathbb{R}^{d_{\text{in}}}$  une fonction sur le graphe et  $\mathbf{g} : V \rightarrow \mathbb{R}^{d_{\text{out}}}$  une fonction sur le graphe. Soit  $P \in \mathbb{R}^{n \times n}$  une matrice de permutation qui appartient à l'espace

On nomme  $L$  la transformation non linéaire définie par la couche spectrale sur graphe. On a

$$\mathbf{g}_j = L(\mathbf{P} \mathbf{S} \mathbf{P}^T, \mathbf{P} \mathbf{f}) = \mathbf{P} L(\mathbf{S}, \mathbf{f}), \quad \forall j \in \{1, \dots, d_{\text{out}}\}$$



# Spectral GNN VI

- On a

$$\mathbf{g}_j = L(\mathbf{PSP}^T, \mathbf{Pf}) = \sigma \left( \sum_{i=1}^{d_{\text{in}}} H_{\theta, ij}(\mathbf{PSP}^T)(\mathbf{Pf}_i) \right)$$

En utilisant la proposition précédente on obtient

$$\mathbf{g}_j = L(\mathbf{PSP}^T, \mathbf{Pf}) = \sigma \left( \mathbf{P} \left( \sum_{i=1}^{d_{\text{in}}} H_{\theta, ij}(\mathbf{S}) \mathbf{f}_i \right) \right)$$

Maintenant en utilisant la linéarité de  $\mathbf{P}$  et la nature élément-wise de  $\sigma$  on obtient

$$\mathbf{g}_j = L(\mathbf{PSP}^T, \mathbf{Pf}) = \mathbf{P} \sigma \left( \sum_{i=1}^{d_{\text{in}}} H_{\theta, ij}(\mathbf{S}) \mathbf{f}_i \right) = \mathbf{PL}(\mathbf{S}, \mathbf{f})$$

**Remark:** Comment choisir nos filtres pour que le calcul soit efficace ?

## Spectral GNN VI

**Definition** (Chebyshev graph convolutional layer): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $L \in \mathbb{R}^{n \times n}$  le laplacien sur graphe. Soit  $(\Phi, \Lambda)$  les vecteurs propres et valeurs propres de  $L$ . Soit  $f : V \rightarrow \mathbb{R}^{d_{\text{in}}}$  une fonction sur le graphe et  $g : V \rightarrow \mathbb{R}^{d_{\text{out}}}$  une fonction sur le graphe. On définit une couche de Chebyshev par

$$g_j = \sigma \left( \sum_{i=1}^{d_{\text{in}}} \sum_{k=0}^K \theta_k^{ij} T_k(\tilde{L}) f_i \right), \quad \forall j \in \{1, \dots, d_{\text{out}}\}$$

avec  $\tilde{L} = \frac{2}{\lambda_{\max}} L - I_d$ ,  $T_k$  le polynome de Chebyshev de degré  $k$  défini par la relation de récurrence

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$

et  $\theta_k^{ij} \in \mathbb{R}$  les paramètres appris.

- La relation de récurrence permet de calculer les  $T_k(\tilde{L})f$  de manière efficace sans avoir à stocker toutes les puissances de  $L$ .
- La normalisation par  $\lambda_{\max}$  permet de s'assurer que les valeurs propres de  $\tilde{L}$  sont dans  $[-1, 1]$  ce qui est nécessaire pour la stabilité des polynomes de Chebyshev.

# Spectral GNN VI

**Definition** (Graph convolutional network layer): Soit  $G = [V, E]$  un graphe avec  $|V| = n$  et  $L \in \mathbb{R}^{n \times n}$  le laplacien sur graphe. Soit  $(\Phi, \Lambda)$  les vecteurs propres et valeurs propres de  $L$ . Soit  $f : V \rightarrow \mathbb{R}^{d_{\text{in}}}$  une fonction sur le graphe et  $g : V \rightarrow \mathbb{R}^{d_{\text{out}}}$  une fonction sur le graphe. On définit une couche de GCN par

$$g = \sigma \left( \left( \tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}} W_{\theta} \right) f \right)$$

avec  $\tilde{W} = W + I_d$  la matrice de poids,  $\tilde{D}$  la matrice de degré associée à  $\tilde{W}$  et  $W_{\theta} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$  les paramètres appris.

- Il existent des familles de methodes de ce type avec des filtres rationnelles, nonlinéaires etc
- On peut les coupler avec des méthode de pooling et un-pooling qui peuvent utilisés des informations géométriques ou sur les features.

**Remark:** Ce type de stratégie peut être utilisé sur des domaines généraux. Il faut simplement avoir un laplacien discret défini et convergent sur le domaine avec bord et l'utiliser à la place du graph-Laplacien.