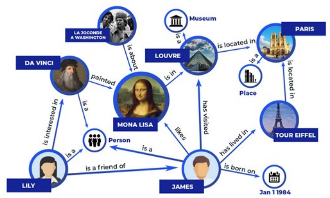
Seeking universal approximation for continuous limits of GNNs on large random graphs

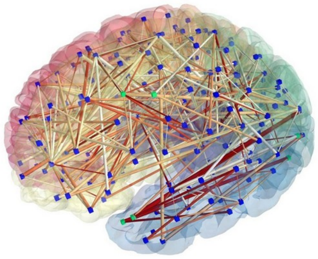
Matthieu Cordonnier[1], **Nicolas Keriven[2]**, Nicolas Tremblay[1], Samuel Vaiter[3]

[1]CNRS, Gipsa-lab, Univ. Grenoble-Alpes
[2]CNRS, IRISA
[3]CNRS, LJAD

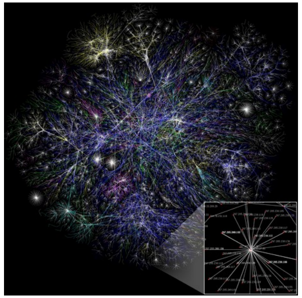
Graphs and large graphs



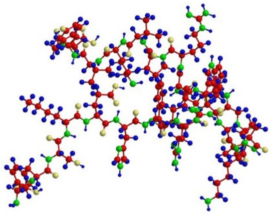
Database/Knowledge graph



Brain connectivity network

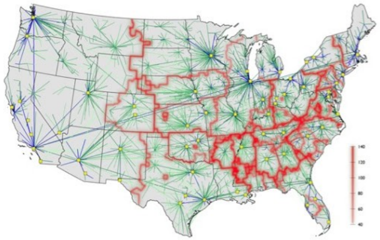


Internet



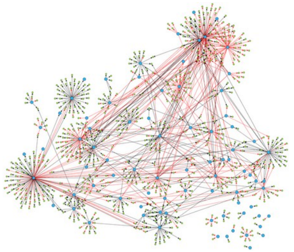
Molecule

Social network

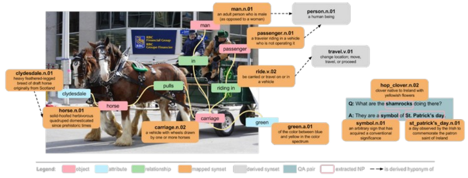


Transportation network

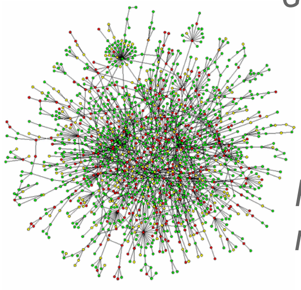
Computer network



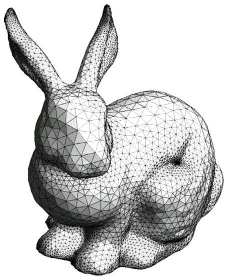
Gene regulatory network



Scene understanding network



Protein interaction network



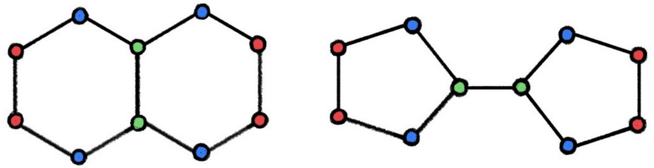
3D mesh

- **Many types of data** can be represented as graphs
  "if all you have is a hammer, everything looks like a nail"
- Many of these graphs are **large**, with **macroscopic properties**
- **GNNs** have become the *de-facto* state-of-the-art models

GNNs and large graphs?

- (Even) compared to regular NNs, many properties of GNNs are still quite mysterious.
  - Eg: **universality** of NNs is known since the 90s, for GNNs it is still a very active field.

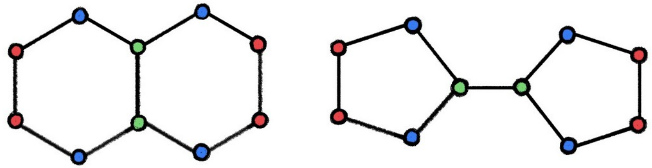
- Most analyses of GNNs are discrete/combinatorial in nature.
  - ***WL-test:*** *Can a GNN distinguish two non-isomorphic graphs?*
  - *Can a GNN count triangles? compute the diameter of a graph? etc.*



GNNs and large graphs?

- (Even) compared to regular NNs, many properties of GNNs are still quite mysterious.
  - Eg: **universality** of NNs is known since the 90s, for GNNs it is still a very active field.

- Most analyses of GNNs are discrete/combinatorial in nature.
  - ***WL-test:*** *Can a GNN distinguish two non-isomorphic graphs?*
  - *Can a GNN count triangles? compute the diameter of a graph? etc.*

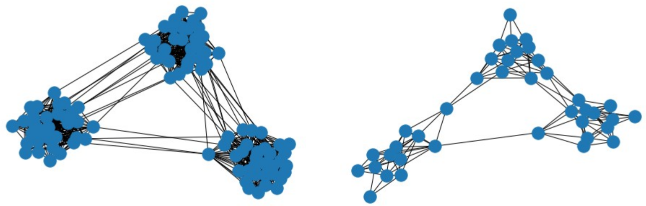


- Large graphs may "look the same", but are never isomorphic, of different size, etc.

- A recent trend is to use **statistical graph models** to understand GNNs' macroscopic behaviors

[see works by Ruiz et al., Levie et al., Keriven et al...]

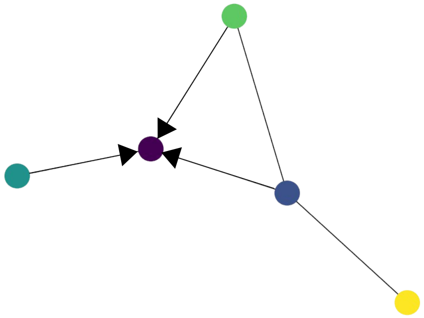
"From discrete to continuous"



GNNs

Most GNNs are based on **message-passing**

$$z_i^{(\ell+1)} = \mathrm{AGG}(z_i^{(\ell)}, \{z_j^{(\ell)}\}_{j\in\mathcal{N}(i)})$$

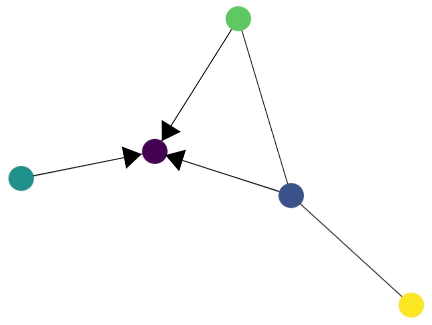


At each layer, each node receives "messages" from its neighbors.

GNNs

Most GNNs are based on **message-passing**

$$z_i^{(\ell+1)} = \mathrm{AGG}(z_i^{(\ell)}, \{z_j^{(\ell)}\}_{j \in \mathcal{N}(i)})$$



At each layer, each node receives "messages" from its neighbors.

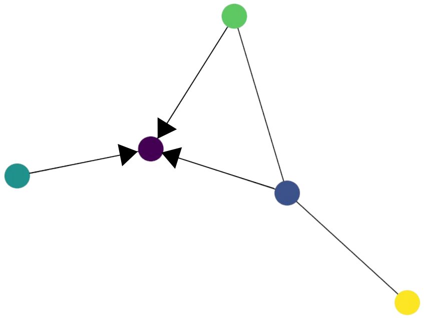
Classical: use of a **graph matrix** S (normalized adjacency, normalized Laplacian...)

$$Z^{(\ell+1)} = \rho\left(Z^{(\ell)}\theta_0^{(\ell)} + S Z^{(\ell)}\theta_1^{(\ell)} + 1_n (b^{(\ell)})^\top\right) \in \mathbb{R}^{n \times d_\ell}$$

GNNs

Most GNNs are based on **message-passing**

$$z_i^{(\ell+1)} = \mathrm{AGG}(z_i^{(\ell)}, \{z_j^{(\ell)}\}_{j\in\mathcal{N}(i)})$$



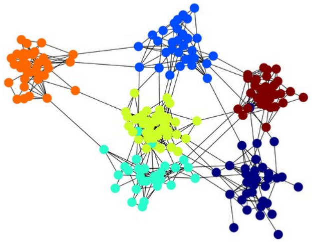
At each layer, each node receives "messages" from its neighbors.

Classical: use of a **graph matrix** S (normalized adjacency, normalized Laplacian...)

$$Z^{(\ell+1)} = \rho\left(Z^{(\ell)}\theta_0^{(\ell)} + SZ^{(\ell)}\theta_1^{(\ell)} + 1_n(b^{(\ell)})^\top\right) \in \mathbb{R}^{n\times d_\ell}$$

Output **node labels:**

$$\Phi_\theta(S, Z^{(0)}) = Z^{(L)}\theta^{(L)} + 1_n(b^{(L)})^\top$$



Can also perform "global pooling" to compute whole graph quantity

$$\frac{1}{n}\sum_i Z_i^{(\ell)}$$

(Example of most) random graphs

Long history of modelling large graphs with **random generative models**

Chung and Lu. *Complex Graphs and Networks* (2004)
Penrose. *Random Geometric Graphs* (2008)
Lovasz. *Large networks and graph limits* (2012)
Frieze and Karonski. *Introduction to random graphs* (2016)

(Example of most) random graphs

Long history of modelling large graphs with **random generative models**

Chung and Lu. *Complex Graphs and Networks* (2004)
Penrose. *Random Geometric Graphs* (2008)
Lovasz. *Large networks and graph limits* (2012)
Frieze and Karonski. *Introduction to random graphs* (2016)

Latent position models *(W-random graphs, kernel random graphs, graphon...)*

$$x_1, \ldots, x_n \overset{iid}{\sim} P_x \qquad a_{ij} \sim P_e(\cdot \mid x_i, x_j)$$

***Unknown** latent variables in a **latent space** $\mathcal{X}$*

*(most often) **independent Edges***

(Example of most) random graphs

Long history of modelling large graphs with **random generative models**

Chung and Lu. *Complex Graphs and Networks* (2004)
Penrose. *Random Geometric Graphs* (2008)
Lovasz. *Large networks and graph limits* (2012)
Frieze and Karonski. *Introduction to random graphs* (2016)

Latent position models *(W-random graphs, kernel random graphs, graphon...)*

$$x_1, \dots, x_n \overset{iid}{\sim} P_x \qquad a_{ij} \sim P_e(\cdot \mid x_i, x_j)$$

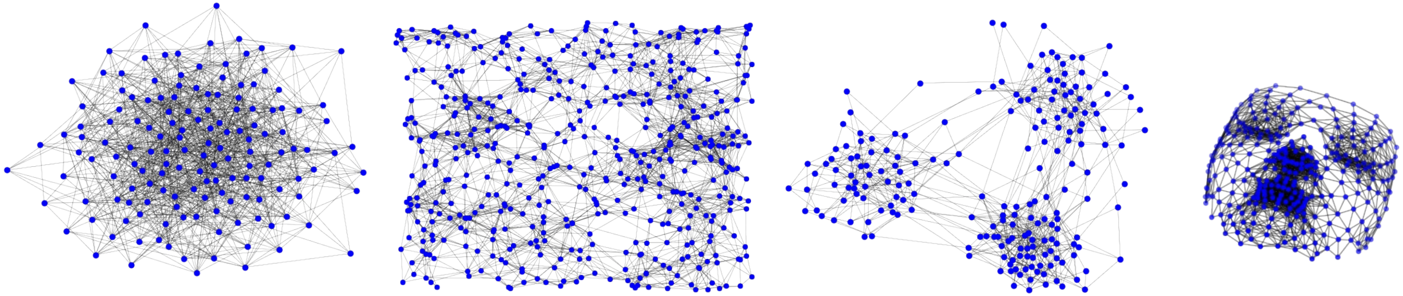
Unknown** latent variables in a **latent space $\mathcal{X}$

*(most often) **independent Edges***

Ex:

$$a_{ij} \sim \mathrm{Ber}(W_n(x_i, x_j))$$

W_n may vary with n to model **graph sparsity**



Includes Erdös-Rényi, Stochastic Block Models, Gaussian kernel, epsilon-graphs...

Continuous GNNs

In many cases, S will "converge" to an **operator** $\mathbf{S}$ (ex: adjacency to kernel integral operator T_W)
One can then define a **continuous GNN** that propagates **functions over the latent space**:

$$f^{(\ell+1)} = \rho\left((\theta_0^{(\ell)})^\top f^{(\ell)} + (\theta_1^{(\ell)})^\top \mathbf{S} f^{(\ell)} + b^{(\ell)}\right) \in L^2(\mathbb{R}^{d_\ell}, P)$$

$$\Phi_\theta(\mathbf{S}, f^{(0)}) = (\theta^{(L)})^\top f^{(L)} + b^{(L)}$$

Global pooling: $\int f^{(\ell)}(x)dP(x)$

Continuous GNNs

In many cases, S will "converge" to an **operator** $\mathbf{S}$ (ex: adjacency to kernel integral operator T_W)

One can then define a **continuous GNN** that propagates **functions over the latent space**:

$$f^{(\ell+1)} = \rho\left((\theta_0^{(\ell)})^\top f^{(\ell)} + (\theta_1^{(\ell)})^\top \mathbf{S} f^{(\ell)} + b^{(\ell)}\right) \in L^2(\mathbb{R}^{d_\ell}, P)$$

$$\Phi_\theta(\mathbf{S}, f^{(0)}) = (\theta^{(L)})^\top f^{(L)} + b^{(L)}$$

Global pooling: $\int f^{(\ell)}(x) dP(x)$

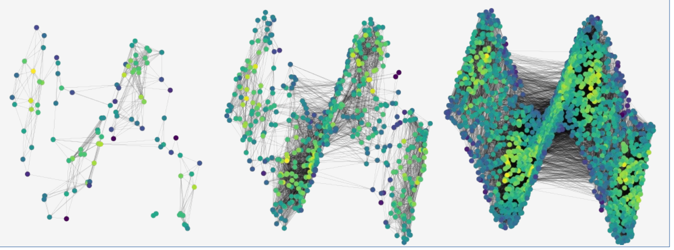
Example of convergence result:

$$\|\cdot\|_{\mathrm{MSE}}^2 = \frac{1}{n}\|\cdot\|_F^2, \ \iota_X f = [f(x_i)]_{i=1}^n$$

If, for all f: $\|S\iota_X f - \iota_X \mathbf{S} f\|_{\mathrm{MSE}} \xrightarrow[n\to\infty]{\mathcal{P}} 0$

Then: $\|\Phi_\theta(S, \iota_X f^{(0)}) - \iota_X \Phi_\theta(\mathbf{S}, f^{(0)})\|_{\mathrm{MSE}} \xrightarrow[n\to\infty]{\mathcal{P}} 0$

GNN applied to a sampled function converge to the sampled output of the cGNN



Valid for adjacency, Laplacian, degree-normalized versions...

Expressivity of cGNNs

$$\Phi_\theta(\mathbf{S}, \cdot) : L^2(\mathcal{X}, P) \to L^2(\mathcal{X}, P)$$

Expressivity of cGNNs have been examined a few times:

- *[Keriven, Vaiter. NeurIPS 2023]*: for a fixed input signal, **range of output functions**
- *[Boker et al. NeurIPS 2023]*: extension of **WL-test to graphons** (ie for graph-tasks)

Expressivity of cGNNs

$$\Phi_\theta(\mathbf{S}, \cdot) : L^2(\mathcal{X}, P) \to L^2(\mathcal{X}, P)$$

Expressivity of cGNNs have been examined a few times:

- *[Keriven, Vaiter. NeurIPS 2023]*: for a fixed input signal, **range of output functions**
- *[Boker et al. NeurIPS 2023]*: extension of **WL-test to graphons** (ie for graph-tasks)

Here: universality of cGNNs as **nonlinear operators** between functions (fixed graph model)

Expressivity of cGNNs

$$\Phi_\theta(\mathbf{S}, \cdot) : L^2(\mathcal{X}, P) \to L^2(\mathcal{X}, P)$$

Expressivity of cGNNs have been examined a few times:

- *[Keriven, Vaiter. NeurIPS 2023]*: for a fixed input signal, **range of output functions**
- *[Boker et al. NeurIPS 2023]*: extension of **WL-test to graphons** (ie for graph-tasks)

Here: universality of cGNNs as **nonlinear operators** between functions (fixed graph model)

Def: Universality

For all: $\mathcal{K} \subset L^2$ compact, $\mathcal{T}$ operator, $\epsilon > 0$

There exists θ such that

$$\sup_{f \in \mathcal{K}} \|\Phi_\theta(f) - \mathcal{T}(f)\|_{L^2} \leq \epsilon$$

Neural Operators

This is reminiscent of **Neural Operators** *[Tianping & Hong 1995, Lu et al. 2021]*

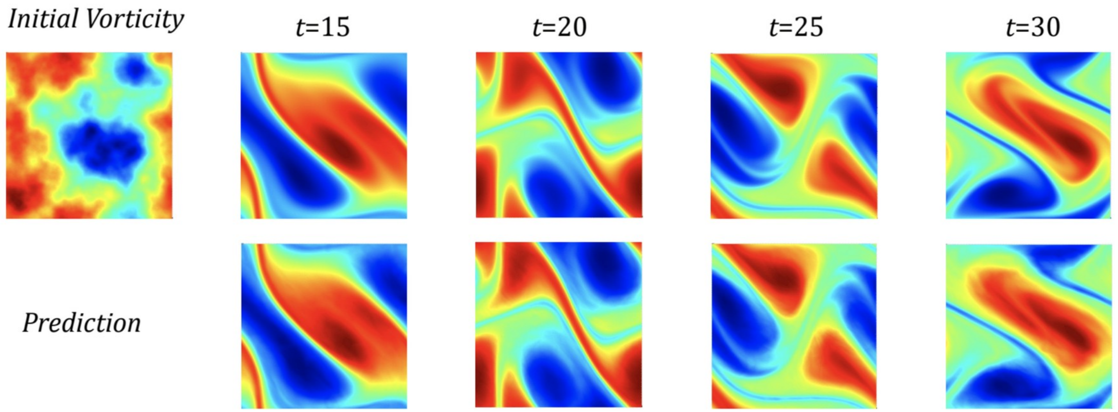
$$\Phi_\theta \approx \mathcal{T} : \mathcal{A} \to \mathcal{B}$$

Neural Operators

This is reminiscent of **Neural Operators** *[Tianping & Hong 1995, Lu et al. 2021]*

$$\Phi_\theta \approx \mathcal{T} : \mathcal{A} \to \mathcal{B}$$

- Used a lot to solve PDEs



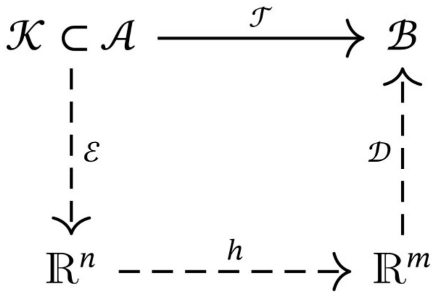
[Li et al. 2021]

- Universality known since the 90s! *[Tianping & Hong 1995]*

Universal Neural Operators? 1/2

How does one construct universal NOs?

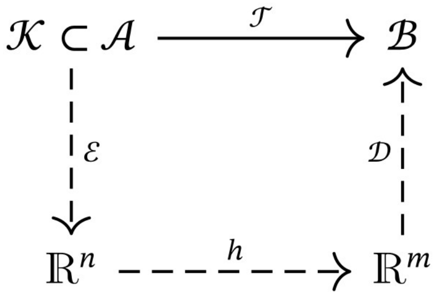
1) Encoder $\mathcal{E}$ into a finite-dimensional space
2) Approximate a continuous mapping h with an MLP
3) Decoder $\mathcal{D}$ from another finite-dimensional space



Universal Neural Operators? 1/2

How does one construct universal NOs?

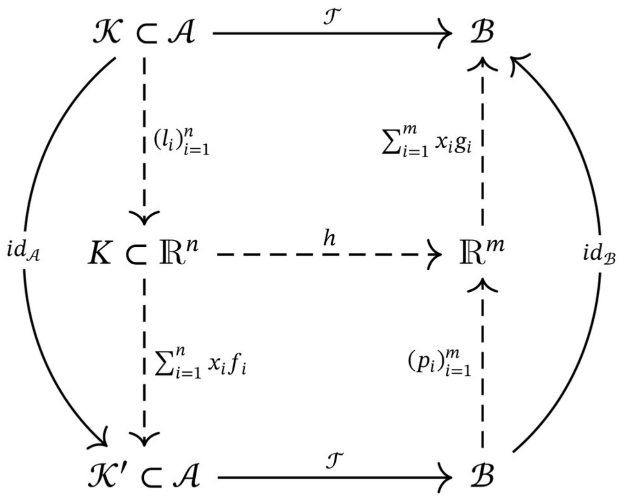
1) Encoder $\mathcal{E}$ into a finite-dimensional space
2) Approximate a continuous mapping h with an MLP
3) Decoder $\mathcal{D}$ from another finite-dimensional space



When is this possible?

When one can construct ***finite-rank operators*** converging to ***identity***

→ so-called *approximation property* of Banach spaces

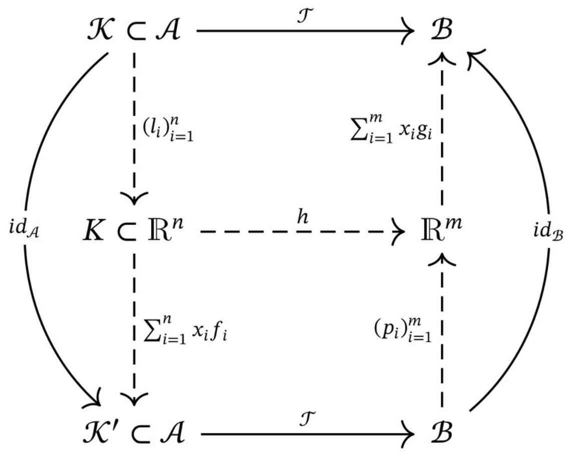


Universal Neural Operators? 2/2

For instance, in Hilbert spaces $\mathcal{A}, \mathcal{B}$, with bases $\{\phi_i\}, \{\psi_j\}$

$$\sum_{j=1}^{m} \mathrm{MLP}_{\theta_j} \left([\langle f, \phi_i \rangle]_{i=1}^{m} \right) \psi_j$$

is universal



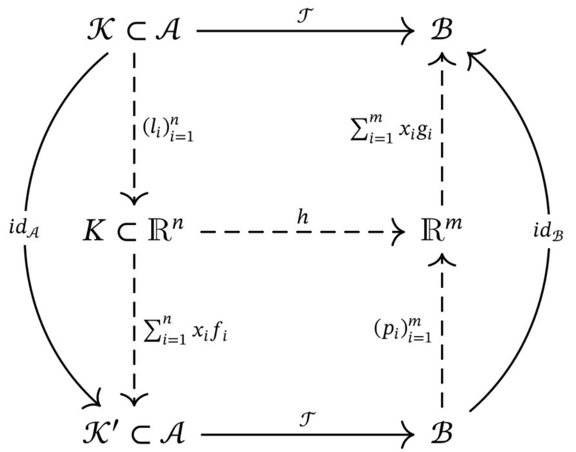
Universal Neural Operators? 2/2

For instance, in Hilbert spaces $\mathcal{A}, \mathcal{B}$, with bases $\{\phi_i\}, \{\psi_j\}$

$$\sum_{j=1}^{m} \mathrm{MLP}_{\theta_j}\left([\langle f, \phi_i\rangle]_{i=1}^{m}\right)\psi_j$$

is universal

→ In practice, ϕ_i, ψ_j are also approximated with MLPs!



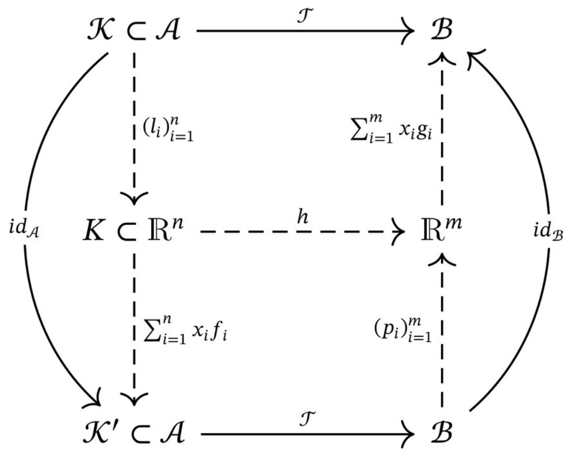
Universal Neural Operators? 2/2

For instance, in Hilbert spaces $\mathcal{A}, \mathcal{B}$, with bases $\{\phi_i\}, \{\psi_j\}$

$$\sum_{j=1}^m \mathrm{MLP}_{\theta_j}\left([\langle f, \phi_i \rangle]_{i=1}^m\right) \psi_j$$

is universal

→ In practice, ϕ_i, ψ_j are also approximated with MLPs!



Can we do this with cGNNs?

→ No! **We don't know the latent variables x_i , so we cannot explicitely compute MLPs in the latent space**

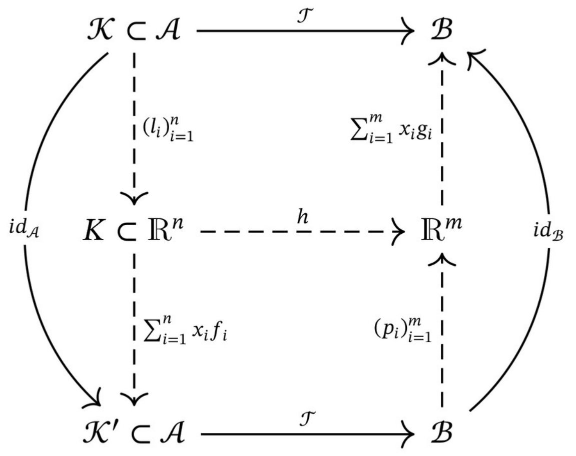
Universal Neural Operators? 2/2

For instance, in Hilbert spaces $\mathcal{A}, \mathcal{B}$, with bases $\{\phi_i\}, \{\psi_j\}$

$$\sum_{j=1}^{m} \mathrm{MLP}_{\theta_j}\left([\langle f, \phi_i \rangle]_{i=1}^{m}\right) \psi_j$$

is universal

→ In practice, ϕ_i, ψ_j are also approximated with MLPs!



Can we do this with cGNNs?

→ No! **We don't know the latent variables x_i, so we cannot explicitely compute MLPs in the latent space**

How do we recover a basis of $L^2(\mathcal{X}, P)$ from $\mathbf{S}$, f ?

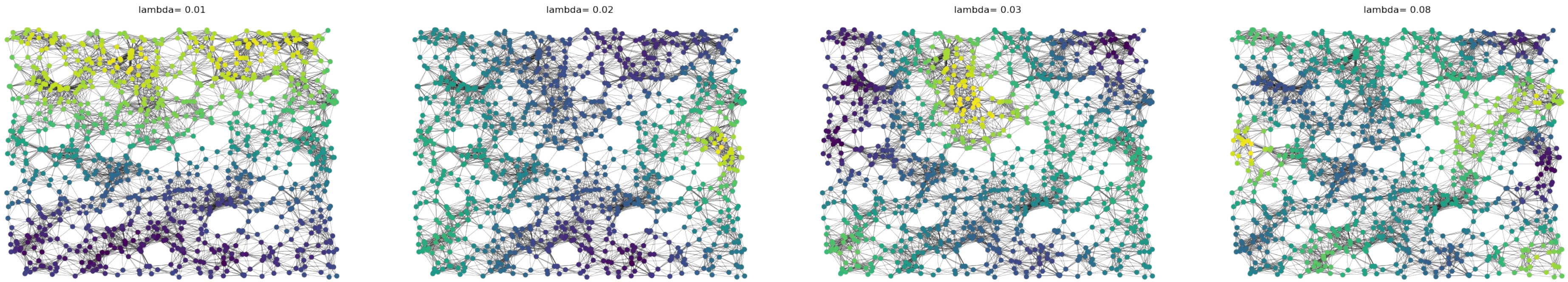
A spectral approach

Assume $\mathbf{S}$ is full-rank → the eigenfunctions $\{\phi_i\}$ are a basis of L2

A spectral approach

Assume $\mathbf{S}$ is full-rank → the eigenfunctions $\{\phi_i\}$ are a basis of L2

Can we just diagonalize $S \approx \mathbf{S}$ *?* → we know $v_i \approx \iota_X \phi_i$



v_1 v_3 v_5 v_{10}

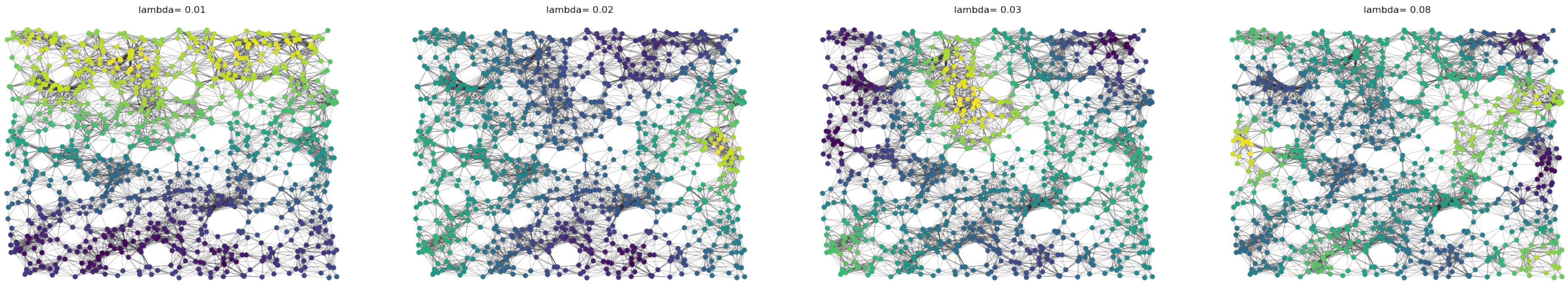
A spectral approach

Assume $\mathbf{S}$ is full-rank → the eigenfunctions $\{\phi_i\}$ are a basis of L2

Can we just diagonalize $S \approx \mathbf{S}$ *?* → we know $v_i \approx \iota_X \phi_i$

→ technically yes, but...

1) GNNs cannot directly diagonalize (… but not a real pbm: Positional Encoding, or clever filtering)



v_1 v_3 v_5 v_{10}

A spectral approach

Assume $\mathbf{S}$ is full-rank → the eigenfunctions $\{\phi_i\}$ are a basis of L2

Can we just diagonalize $S \approx \mathbf{S}$ *?* → we know $v_i \approx \iota_X \phi_i$

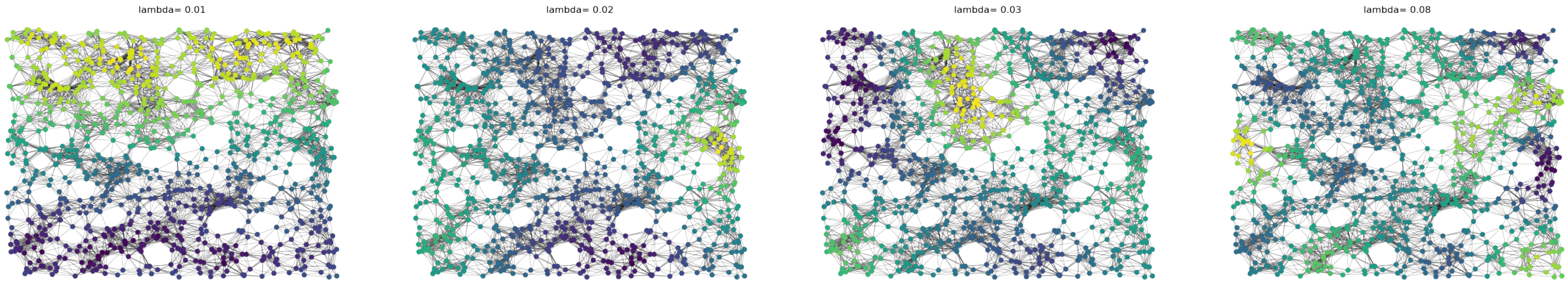
→ technically yes, but...

1) GNNs cannot directly diagonalize (… but not a real pbm: Positional Encoding, or clever filtering)

2) Sign(/basis) **indeterminancy**: cannot generalize!

$$\text{output} = \{\text{random\_sign}_i \phi_i\}$$

at each new graph/diagonalization algorithm



v_1 v_3 v_5 v_{10}

Sign indeterminancy?

- Not a problem for **filtering**:

$$h(\mathbf{S})f = \sum_i h(\lambda_i) \underbrace{\langle f, \phi_i \rangle \phi_i}_{\text{Invariant to sign change!}}$$

Sign indeterminancy?

- Not a problem for **filtering**:

$$h(\mathbf{S})f = \sum_i h(\lambda_i) \underbrace{\langle f, \phi_i \rangle \phi_i}_{\text{Invariant to sign change!}}$$

- Very much a problem for us!

$$\textstyle\sum_{j=1}^{m} \mathrm{MLP}_{\theta_j} \left([\langle f, \phi_i \rangle]_{i=1}^{m} \right) \phi_j$$

$\mathrm{MLP} : \mathbb{R}^m \to \mathbb{R}$ *has no reason to be sign invariant for each coordinate!*

Sign indeterminancy: quick fix

Assume $\mathbf{S}$ is full-rank (with simple eigenvalues)

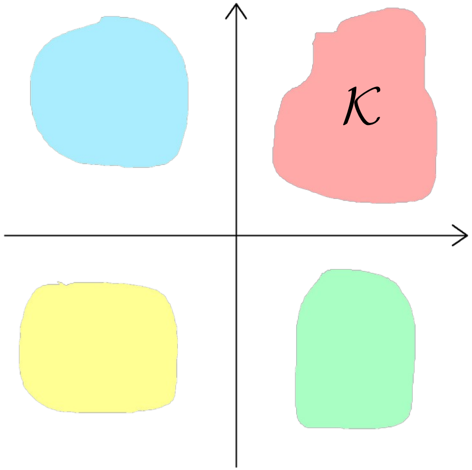
We are working **within a compact of functions** anyway, let's assume that it is in a **fixed orthant**

Assume: $\mathcal{K} \subset \mathcal{D}^+$

$\mathcal{D}^+$ s.t. there exists $\{\phi_i\}$ basis of eigenfunctions of $\mathbf{S}$ s.t.

$$\forall f \in \mathcal{D}^+, i \geq 1 \qquad \langle f, \phi_i \rangle_{L^2} > 0$$

w.l.o.g.



A spectral NO

A spectral approach:

$$\Phi_\theta(\mathbf{S}, f) =$$

1) Diagonalize $\mathbf{S} \to \phi_1, \ldots, \phi_m$

2) Solve sign indeterminancy by $\phi_i \leftarrow \text{sign}(\langle f, \phi_i \rangle)\phi_i$

3) Return $\sum_{j=1}^{m} \text{MLP}_{\theta_j}\left([\langle f, \phi_i \rangle]_{i=1}^{m}\right)\phi_j$

A spectral NO

A spectral approach:

$$\Phi_\theta(\mathbf{S}, f) =$$

1) Diagonalize $\mathbf{S} \rightarrow \phi_1, \ldots, \phi_m$

2) Solve sign indeterminancy by $\phi_i \leftarrow \mathrm{sign}(\langle f, \phi_i \rangle)\phi_i$

3) Return $\sum_{j=1}^m \mathrm{MLP}_{\theta_j}\left([\langle f, \phi_i \rangle]_{i=1}^m\right)\phi_j$

Theorem:

Universal on domains $\mathcal{K} \subset \mathcal{D}^+$

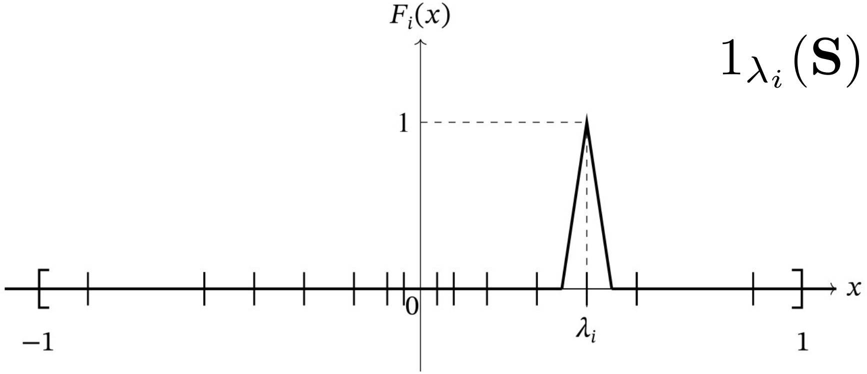
Actual GNNs?

All this spectral stuff is a bit weird. Can we recover the eigenfunctions by actual GNNs?

Actual GNNs?

All this spectral stuff is a bit weird. Can we recover the eigenfunctions by actual GNNs?

Since eigenvalues are separated and $\mathcal{K} \subset \mathcal{D}^+$, ϕ_i can be recovered by filtering and normalization



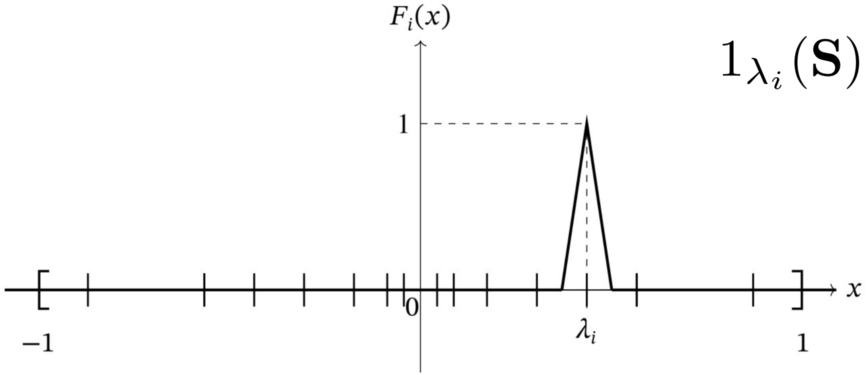
$$1_{\lambda_i}(\mathbf{S})f = \langle f, \phi_i \rangle \phi_i \qquad \|1_{\lambda_i}(\mathbf{S})f\| = |\langle f, \phi_i \rangle| = \langle f, \phi_i \rangle$$

$$\frac{1_{\lambda_i}(\mathbf{S})f}{\|1_{\lambda_i}(\mathbf{S})f\|} = \phi_i$$

Actual GNNs?

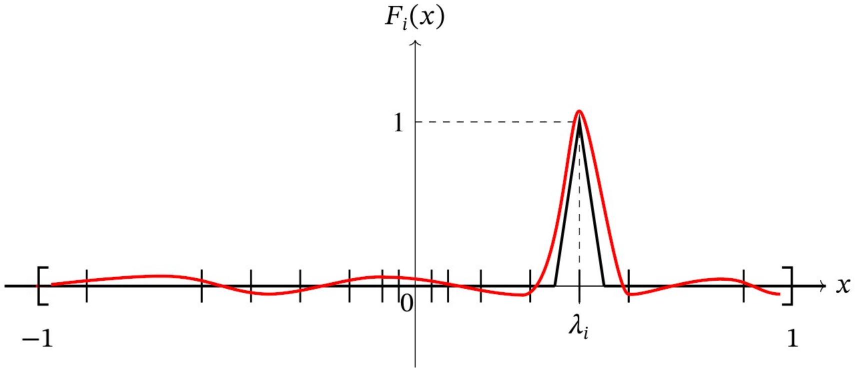
All this spectral stuff is a bit weird. Can we recover the eigenfunctions by actual GNNs?

Since eigenvalues are separated and $\mathcal{K} \subset \mathcal{D}^{+}$, ϕ_i can be recovered by filtering and normalization



$$1_{\lambda_i}(\mathbf{S})f = \langle f, \phi_i \rangle \phi_i \qquad \|1_{\lambda_i}(\mathbf{S})f\| = |\langle f, \phi_i \rangle| = \langle f, \phi_i \rangle$$

$$\frac{1_{\lambda_i}(\mathbf{S})f}{\|1_{\lambda_i}(\mathbf{S})f\|} = \phi_i$$



Since universality is up to ϵ, they can even be replaced by polynomial filters!

$$\frac{Q_i(\mathbf{S})f}{\|Q_i(\mathbf{S})f\|} \approx \phi_i$$

Actual cGNNs

Theorem:

cGNNs are universal on domains $\mathcal{K} \subset \mathcal{D}^+$

Actual cGNNs

Theorem:

cGNNs are universal on domains $\mathcal{K} \subset \mathcal{D}^+$

Proof:

$$\sum_{j=1}^{m} \mathrm{MLP}_{\theta_j}\left([\|Q_i(\mathbf{S})f\|]_{i=1}^{m}\right) \frac{Q_j(\mathbf{S})f}{\|Q_j(\mathbf{S})f\|}$$

are universal

Actual cGNNs

Theorem:

cGNNs are universal on domains $\mathcal{K} \subset \mathcal{D}^+$

Proof:

$$\sum_{j=1}^{m} \mathrm{MLP}_{\theta_j}\left([\|Q_i(\mathbf{S})f\|]_{i=1}^{m}\right) \frac{Q_j(\mathbf{S})f}{\|Q_j(\mathbf{S})f\|}$$

are universal

Nothing more that regular cGNNs!! (with intermediate global poolings)

- **Polynomial filtering**
- **Global pooling** to compute norm/inner products $\|f\|_{L^2}^2 \approx \frac{1}{n}\sum_i f(x_i)^2$
- All operations (multiplication, normalization, square root...) can be **approximated by MLPs**

Conclusion

- “From discrete to continuous” is a fruitful approach for GNNs
- Link with Neural Operators
- Still many things to explore/understand in the continuous world!

Outlooks

- Getting rid of the weird sign assumption?

→ **SignNet**-like architecture *[Lim et al. 2022]*:

$$\langle f, \phi_i \rangle \leftarrow \mathrm{MLP}(\langle f, \phi_i \rangle) + \mathrm{MLP}(-\langle f, \phi_i \rangle)$$

$$\phi_i \leftarrow \mathrm{MLP} \circ \phi_i + \mathrm{MLP} \circ (-\phi_i)$$

Outlooks

- Getting rid of the weird sign assumption?

→ **SignNet**-like architecture *[Lim et al. 2022]*:

$$\langle f, \phi_i \rangle \leftarrow \mathrm{MLP}(\langle f, \phi_i \rangle) + \mathrm{MLP}(-\langle f, \phi_i \rangle)$$
$$\phi_i \leftarrow \mathrm{MLP} \circ \phi_i + \mathrm{MLP} \circ (-\phi_i)$$

- More general **Message-Passing** GNN? (GANs...)

→ ***Convergence*** is known

Non-linear operator filtering!

Cordonnier et al., *Convergence of Message Passing Graph Neural Networks with Generic Aggregation On Large Random Graphs*, 2023

Thank you!

nkeriven.github.io

European Research Council
Established by the European Commission

Come work in the beautiful city of Rennes!