



**HAL**  
open science

# Seeking universal approximation for continuous limits of graph neural networks on large random graphs

Matthieu Cordonnier, Nicolas Keriven, Nicolas Tremblay, Samuel Vaiter

## ► To cite this version:

Matthieu Cordonnier, Nicolas Keriven, Nicolas Tremblay, Samuel Vaiter. Seeking universal approximation for continuous limits of graph neural networks on large random graphs. Asilomar 2024 - Asilomar Conference on Signals, Systems, and Computers, Oct 2024, Pacific Grove, United States. pp.1-5. hal-04831761

**HAL Id: hal-04831761**

**<https://hal.science/hal-04831761v1>**

Submitted on 11 Dec 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Seeking universal approximation for continuous limits of graph neural networks on large random graphs.

Matthieu Cordonnier  
GIPSA-lab  
Université Grenoble Alpes  
Grenoble, France

Nicolas Keriven  
IRISA  
CNRS  
Rennes, France

Nicolas Tremblay  
GIPSA-lab  
CNRS  
Grenoble, France

Samuel Vaiter  
Laboratoire J. A. Dieudonné  
CNRS  
Nice, France

**Abstract**—We propose a notion of universality for graph neural networks (GNNs) in the large random graphs limit, tailored for node-level tasks. When graphs are drawn from a latent space model, or from a graphon, GNNs on growing graph sequences are known to converge to limit objects called “continuous GNNs”, or cGNNs. A cGNN inputs and outputs functions defined on a latent space, and as such, is a non-linear operator between functional spaces. Therefore, we propose to evaluate the expressivity of a cGNN through its ability to approximate arbitrary non-linear operators between functional spaces. This is reminiscent of Operator Learning, a branch of machine learning dedicated to learning non-linear operators between functional spaces. In this field, several architectures known as Neural Operators (NOs) are indeed proven to be universal. The justification for the universality of these architectures relies on a constructive method based on an encoder-decoder strategy into finite dimensional spaces, which enables invoking the universality of usual MLP. In this paper, we adapt this method to cGNNs. This is however far from straightforward: cGNNs have crucial limitations, as they do not have access to the latent space, but observe it only indirectly through the graph built on it. Our efforts will be directed toward circumventing this difficulty, which we will succeed, at this stage, only with strong hypotheses.

**Index Terms**—graph neural networks, random graphs, neural operators.

## I. INTRODUCTION

Despite being state-of-the-art in many practical situations, the properties of graph neural networks (GNNs) remain poorly understood. The dominant approach for theoretical analyses of GNNs is based on combinatorial approaches, mostly via graph isomorphism testing and comparison to the Weisfeiler-Lehman (WL) heuristic [1]. However, this point of view is somewhat of limited interest for large graphs and node-level tasks. Large graphs are better understood through their macroscopic properties and patterns rather than their exact adjacency structure. Moreover, isomorphism testing is inherently graph-level, and bears limited connection to node-level tasks. In general, notions of expressive power for GNNs in the context of node-level tasks are somewhat scarcer. In this study, we propose a notion of expressive power for GNN in the large graphs limit, tailored for node-level tasks.

A recent axis of research make use of statistical models to explore properties of GNNs on large-scale graphs [2]–[4]. In

particular, when graphs are drawn from a *latent space* random graph model [5], spectral GNNs with polynomial filters [3], [6], as well as some message-passing GNNs [7] are known to converge toward some limit architectures as the size of the input graphs tends to infinity. These limit architectures are known as *continuous graph neural networks* (cGNNs), they are the “continuous counterparts” of discrete GNNs. While GNNs process graph signals, cGNNs process *functions defined on the latent space*.

In the literature, limits of GNNs on large random graphs are used to gain insight on generalisation [8], transferability [2], [6], as well as expressivity capabilities of GNNs Keriven *et al.* [9], Keriven and Vaiter [10], and Böker *et al.* [11]. Here, we focus on expressivity aspects. Previous work from Keriven and Vaiter [10] explore the range of output functions that a cGNN can compute for a predetermined input signal, while Böker *et al.* [11] extend the WL test to the continuous setting. In this paper, we propose to consider the limit cGNN as a non-linear operator between functional spaces, and explore the range of non-linear operator that cGNNs can approximate. This is reminiscent of another branch of machine learning called *operator learning* (OL), which will be the starting point of our theoretical analysis. Technical proofs are available at Cordonnier [12].

## II. PRELIMINARIES AND PROBLEM STATEMENT

### A. Latent position random graph model

The *latent space* is a probability space  $(\mathcal{X}, P)$ , where  $\mathcal{X} \subset \mathbf{R}^d$  is assumed to be a compact subset of a potentially high-dimensional real vector space. The latent space model is a triplet  $(\mathcal{X}, P, W)$  where  $W: \mathcal{X}^2 \rightarrow [0, 1]$  is a connectivity kernel. A random graph drawn of size  $n$  from the model has *unobserved* random latent positions  $X_1, \dots, X_n \in \mathcal{X}$  as vertices, and a random adjacency matrix  $A = (a_{i,j})$ . The random variables  $X_i$  and  $a_{i,j}$  are distributed according to:

$$X_1, \dots, X_n \stackrel{\text{i.i.d.}}{\sim} P \quad \text{and} \quad a_{i,j} \stackrel{\text{i.i.d.}}{\sim} \mathcal{Ber}(W(X_i, X_j)) .$$

The latent position model encompasses many important model such as the stochastic block model [13], as well as Erdős-Rényi, and geometric random graph models [14], [15]. Notice that,

by definition, the latent positions of the node are *unobserved* and that the latent space is *inaccessible*.

### B. Graph neural network and graph neural network limits

1) *Graph neural networks*: GNNs are based on the *message passing* [16] paradigm in which the features of each node are sequentially updated by aggregating the features of its neighbours. Classically, this is implemented using a matrix representative of the graph [17], such as the adjacency matrix or the graph Laplacian, with various normalizations. Let  $G_n$  be a graph of size  $n$  and  $S$  be a graph matrix representative, and  $Z = Z^{(0)} \in \mathbf{R}^{n \times d_0}$  be an input graph signal. A traditional GNN sequentially update the graph signal as follows:

$$Z^{(l+1)} = \rho \left( Z^{(l)} \theta_0^{(l)} + S Z^{(l)} \theta_1^{(l)} + 1_n \left( b^{(l)} \right)^\top \right), \quad (1)$$

where  $\rho$  is a point-wise activation function such as ReLU,  $1_n$  is the all-ones vector, and the  $\theta_i^{(l)} \in \mathbf{R}^{d_l \times d_{l+1}}$ ,  $b^{(l)} \in \mathbf{R}^{d_l}$  are the learnable parameters. The output of the GNN is  $\Phi(S, Z) = Z^{(L)} \in \mathbf{R}^{n \times d_L}$ . Additionally, we assume that GNNs can compute global pooled quantity for the graph:  $A^{(l)} = 1_n^\top Z^{(l)} / n$ , that can be transmitted as a new node feature (equal for all nodes) in subsequent layers.

2) *Limits of graph neural networks on large random graphs*: When graphs are drawn from a latent space model  $(\mathcal{X}, P, W)$ , there are several cases where a matrix representative  $S$  is known to converge to an operator  $\mathcal{S}$  [10]. For example,  $S$  can be the normalized random adjacency matrix  $A/n$  and  $\mathcal{S}$  the integral operator

$$T_W : \begin{cases} \mathbf{L}^2(\mathcal{X}, P) & \longrightarrow & \mathbf{L}^2(\mathcal{X}, P) \\ f & \longmapsto & \int W(x, \cdot) f(x) dP(x). \end{cases} \quad (2)$$

Consequently, it is natural to define a continuous counterpart of the GNN  $\Phi(S, \cdot)$ . We call it a continuous GNN (cGNN) and denote it by  $\Phi(\mathcal{S}, \cdot)$ . Rather than processing discrete signals on a random graph, the cGNN processes functions on the latent space  $\mathcal{X}$ . The continuous counterpart of  $\Phi(S, \cdot)$  shares the same set of learnable parameters from Equation (1). Starting from an input function  $f = f^{(0)} \in \mathbf{L}^2(\mathcal{X}, P)$ , the layer-wise update rule is

$$f^{(l+1)} = \rho \left( \left( \theta_0^{(l)} \right)^\top f^{(l)} + \left( \theta_1^{(l)} \right)^\top \mathcal{S} f^{(l)} + b^{(l)} \right), \quad (3)$$

where  $\rho$ , the  $\theta_i^{(l)}$ , and the  $b^{(l)}$ , are the same as in Equation (1). The output is then  $f^{(L)}$ . Finally, global pooling here correspond to the integral:  $A^{(l)} = \int f^{(l)} dP$ .

Several authors [3], [4], [7] have shown that, in many cases, the discrete GNN  $\Phi(S, \cdot)$  on random growing graphs converges to its continuous cGNN counterpart  $\Phi(\mathcal{S}, \cdot)$ . This convergence is described through a sampling procedure. Let  $\iota_X$  denote the sampling operator at the random latent positions  $X = (X_1, \dots, X_n)$ , that is

$$\iota_X f = (f(X_1), \dots, f(X_n)).$$

Then,  $\Phi(S, \cdot)$  is said to converge to  $\Phi(\mathcal{S}, \cdot)$  if

$$\|\Phi(S, \iota_X f) - \iota_X \Phi(\mathcal{S}, f)\| \xrightarrow{n \rightarrow \infty} 0 \quad (4)$$

in probability for a certain class of functions  $f$  in  $\mathbf{L}^2(\mathcal{X}, P)$  and a certain norm  $\|\cdot\|$ .

### C. Expressivity of continuous graph neural networks as non-linear operators.

A GNN is a non-linear operator which inputs a signal on a graph and outputs another signal on the same graph. Therefore, a natural notion of expressive power can be the following: what is the range of non-linear transformations between graph signals that can be approximated by GNN? This is difficult to define because the graph – and consequently the signal – can be of any size. Hence, instead of considering the discrete GNN directly, we are going to focus on its continuous counterpart, the cGNN. We will consider expressivity based on usual universal approximation property, adapted to our context, which definition is given below.

**Definition 1** (Universal Approximation (UA) Property.). *Let  $\mathcal{H}$  be a Hilbert space, and  $\mathcal{G} \subset \mathcal{C}(\mathcal{H}, \mathcal{H})$  be a family of continuous maps from  $\mathcal{H}$  to itself. We say that  $\mathcal{G}$  has the universal approximation (UA) property if the following is satisfied. For any  $\mathcal{K} \subset \mathcal{H}$  compact, for any  $\mathcal{T} \in \mathcal{C}(\mathcal{H}, \mathcal{H})$ , for any  $\varepsilon > 0$ , there exists a  $\Psi \in \mathcal{G}$  such that*

$$\sup_{f \in \mathcal{K}} \|\Psi(f) - \mathcal{T}(f)\|_{\mathcal{H}} \leq \varepsilon. \quad (5)$$

In the case where  $\Phi(S, \cdot)$  is a GNN which tends to a continuous counterpart  $\Phi(\mathcal{S}, \cdot)$  as in Equation (4), we have the approximation  $\Phi(S, \cdot) \circ \iota_X \approx \iota_X \circ \Phi(\mathcal{S}, \cdot)$  for large random graphs. Hence, we expect that the expressivity of  $\Phi(\mathcal{S}, \cdot)$  will be a good estimation of the one of  $\Phi(S, \cdot)$ .

## III. OPERATOR LEARNING

### A. A brief review of operator learning

The branch of machine learning devoted to learn unknown operators between infinite dimensional function spaces is known as operator learning. It is a recent and active topic which aims to extend deep learning methods from finite to infinite dimensional settings. Given a dataset of pairs  $(f, g)$ , where  $f \in \mathcal{U}$  and  $g \in \mathcal{V}$  belong to some function space, the goal is to learn a non-linear operator  $\mathcal{T}: \mathcal{U} \rightarrow \mathcal{V}$  such that  $\mathcal{T}(f) = g$ . This is especially relevant in the context partial differential equations (PDE), for learning the operator that gives the solutions of a PDE. Can we, in a deep learning fashion, learn the non-linear operator that inputs coefficients (usually non constant, therefore functions) and outputs a solution of a PDE? The modern types of deep learning architectures used in this context are called neural operators (NOs) [18]. Our interest into NO lies in the fact that universal approximation theorems do exist for them. Hence, we will try to adapt the methods for proving UA from the context of NO to our problem of cGNN.

### B. Universal neural operators

Neural operators' power strongly rely on the universality of MLP between finite dimensional vector spaces. Therefore, the design of universal NO architecture are based on an encoder-decoder decomposition through finite dimensional spaces. Let

$\mathcal{H}$  be a Hilbert space and  $\mathcal{T} \in \mathcal{C}(\mathcal{H}, \mathcal{H})$ , Kovachki *et al.* [18] proves that  $\mathcal{T}$  can be approximately factorised via the following *encoder/decoder* method. Let  $\varepsilon > 0$  and  $\mathcal{K} \subset \mathcal{H}$  compact, there is  $n \in \mathbf{N}^*$  and three continuous maps  $\mathcal{E} : \mathcal{H} \rightarrow \mathbf{R}^n$ ,  $h : \mathbf{R}^n \rightarrow \mathbf{R}^n$  and  $\mathcal{D} : \mathbf{R}^n \rightarrow \mathcal{H}$  such that :

$$\sup_{f \in \mathcal{K}} \|\mathcal{T}(f) - \mathcal{D} \circ h \circ \mathcal{E}(f)\|_{\mathcal{B}} < \varepsilon. \quad (6)$$

Let  $(\varphi_i)$  be an orthonormal basis of  $\mathcal{H}$ . It turns out that the encoder  $\mathcal{E}$  and the decoder  $\mathcal{D}$  can be chosen as the projection on the truncated basis

$$\mathcal{E} : \begin{cases} \mathcal{H} & \rightarrow \mathbf{R}^n \\ f & \mapsto (\langle f, \varphi_1 \rangle, \dots, \langle f, \varphi_n \rangle), \end{cases}$$

and the expansion

$$\mathcal{D} : \begin{cases} \mathbf{R}^n & \rightarrow \mathcal{H} \\ x & \mapsto \sum_{i=1}^n x_i \varphi_i. \end{cases}$$

Moreover, since  $h$  is a continuous map between finite dimensional spaces, it can be approximated by a MLP  $h^{\text{MLP}} \approx h$ .

Equation (6) can be summed up using the following approximately commutative diagram, where the dashed path in an approximation of the plain path.

$$\begin{array}{ccc} \mathcal{K} \subset \mathcal{H} & \xrightarrow{\mathcal{T}} & \mathcal{H} \\ \downarrow \text{dashed} & & \uparrow \text{dashed} \\ K \subset \mathbf{R}^n & \xrightarrow{h} & \mathbf{R}^n \\ \downarrow \text{dashed} & & \uparrow \text{dashed} \\ \mathcal{K}' \subset \mathcal{H} & \xrightarrow{\mathcal{T}} & \mathcal{H} \end{array} \quad (7)$$

$\text{dashed paths: } \mathcal{K} \subset \mathcal{H} \xrightarrow{(\langle \cdot, \varphi_i \rangle)_{i=1}^n} K \subset \mathbf{R}^n \xrightarrow{\sum_{i=1}^n x_i \varphi_i} \mathcal{H}$   
 $\mathcal{H} \xrightarrow{\sum_{i=1}^n x_i \varphi_i} \mathbf{R}^n \xrightarrow{(\langle \cdot, \varphi_i \rangle)_{i=1}^n} \mathcal{K}' \subset \mathcal{H}$

This decomposition can be extended to more general Banach spaces having the so-called *approximation property*, that is, in which the identity map can be approximated by finite rank linear operators on compact subsets.

Overall, the following family has the UA property from Definition 1:

$$\left\{ f \mapsto \sum_{i=1}^n h^{\text{MLP}}([\langle f, \varphi_i \rangle]_{i=1}^n) \varphi_i \right\}, \quad (8)$$

where  $n$  ranges over  $\mathbf{N}^*$  and  $h^{\text{MLP}}$  ranges over MLPs from  $\mathbf{R}^n$  into itself. This is actually not a difficult problem. Considering that an orthonormal basis is usually unknown, the last step to get universal NOs is to *learn* the  $\varphi_i$ . This is not difficult, since the  $\varphi_i$  are simply some function on the domain of a PDE, which is known, one can replace them by MLPs directly parametrized on the domain.

This last point is important because, until then, everything was transportable to our cGNN approximation setting. However, replacing the  $\varphi_i$  by MLPs is not possible in our cGNN settings, because the counterpart of the domain of the PDE is the latent space  $\mathcal{X}$ . The latent space  $\mathcal{X}$  is by essence unknown

inaccessible, **it is impossible to train any MLP defined on  $\mathcal{X}$** . Instead, cGNNs only have access to the operator  $\mathcal{S}$ . However, if  $\mathcal{S}$  is full-rank, its basis of eigenvectors is indeed a basis for  $L^2$ . This is the strategy we exploit in the rest of the paper.

#### IV. HILBERT SPACE AND SELF-ADJOINT KERNEL AND CONTINUOUS GRAPH NEURAL NETWORKS

Back to our setting,  $\mathcal{H} = L^2(\mathcal{X}, P)$ . We suppose that the operator  $\mathcal{S}$  is self-adjoint and compact, for example when  $\mathcal{S} = T_W$  for  $W \in L^2(\mathcal{X}^2, P \otimes P)$  and symmetrical. In this section,  $(\varphi_i)_{i \geq 0}$  is an orthonormal Hilbert basis or eigenfunctions of  $\mathcal{S}$  with regard to the eigenvalues  $(\lambda_i)_{i \geq 0}$ . We assume that the eigenvalues are all non-zero and have no multiplicity greater than one.

Our goal is to adapt the universal family from Equation (8) to get a family of cGNNs. However, the eigenfunctions  $\varphi_i$  are unknown, and the question is to know if cGNNs can recover them, or at least approximate them, without knowledge of the latent space  $\mathcal{X}$ ? Recall that among cGNNs' operation there are filtering, composition by a MLP, global pooling, and so on.

##### A. Ambiguity of sign and orthant assumption

Observing Equation (8), notice that this universal family depends on the choice of an orthonormal eigenbasis of  $\mathcal{S}$ . Indeed, under our assumptions, the eigenbasis of  $\mathcal{S}$  is defined **up to sign flip of each eigenfunctions**. In other words, when diagonalizing  $\mathcal{S}$ , one gets the  $\{\phi_i\}$  with random sign flips that might change each time, which prevent generalization. In particular, the expression

$$\sum_{i=1}^n h^{\text{MLP}}([\langle f, \varphi_i \rangle]_{i=1}^n) \varphi_i,$$

is *not* invariant to sign flip of the  $\varphi_i$ , due to the composition by the MLP  $h^{\text{MLP}}$ . Remark that, in contrast, usual linear filtering is always independent from any choice of the eigenbasis, since  $f \mapsto \mathcal{S}f = \sum_{i=1}^n \langle f, \varphi_i \rangle \varphi_i$  is naturally invariant to sign flip of the  $\varphi_i$ . But here we are indeed aiming to perform non-linear filtering.

To resolve this fundamental problem of sign ambiguity, we make the strong assumption to restrict our input compact set to an orthant of the basis  $\{\phi_i\}$ . Without loss of generality, since the  $\phi_i$  are defined up their signs, we consider the *positive* orthant:

$$\mathcal{O}^+ = \{f \in \mathcal{H} \mid \langle f, \varphi_i \rangle > 0 \ \forall i \in \mathbf{N}^*\}. \quad (9)$$

We will then assume that  $\mathcal{K} \subset \mathcal{O}^+$ .

##### B. Approximation of the eigenfunctions by filtering

From now, we restrict to compact sets  $\mathcal{K} \subset \mathcal{O}^+$ . For all eigenvalue of  $\mathcal{S}$ , denote  $\mathbf{1}_{\lambda_i}$  the indicator function of  $\lambda_i$ . It is the function that maps  $\lambda_i$  to 1 and  $\lambda_j$  to 0 if  $i \neq j$ . Notice that, in virtue of the continuous functional calculus of compact self-adjoint operator, the filtering  $\mathbf{1}_{\lambda_i}(\mathcal{S})$  is the orthoprojector on the eigenspace  $\ker(\mathcal{S} - \lambda_i \text{id})$ . Hence since we assume this eigenspace to be one-dimensional, this yields

$$\mathbf{1}_{\lambda_i}(\mathcal{S})f = \langle f, \varphi_i \rangle \varphi_i. \quad (10)$$

Moreover, since  $\varphi_i$  is unitary, and from Equation (9), for any  $f \in \mathcal{K} \subset \mathcal{O}^+$ ,

$$\|\mathbf{1}_{\lambda_i}(\mathcal{S})(f)\| = |\langle f, \varphi_i \rangle| = \langle f, \varphi_i \rangle > 0. \quad (11)$$

As a consequence, we can define on  $\mathcal{K} \subset \mathcal{O}^+$ :

$$\frac{\mathbf{1}_{\lambda_i}(\mathcal{S})(f)}{\|\mathbf{1}_{\lambda_i}(\mathcal{S})(f)\|} = \text{sgn}(\langle f, \varphi_i \rangle) \varphi_i = \varphi_i. \quad (12)$$

1) *Perfect eigenprojector filters*: As a result from Equations (10) to (12), the following family is universal on compacts  $\mathcal{K} \subset \mathcal{O}^+$ .

$$\left\{ f \mapsto \sum_{i=1}^n h^{\text{MLP}} \left( \left( \|\mathbf{1}_{\lambda_i}(\mathcal{S})f\|_{i=1}^n \right) \frac{\mathbf{1}_{\lambda_i}(\mathcal{S})f}{\|\mathbf{1}_{\lambda_i}(\mathcal{S})f\|} \right) \right\}.$$

In addition, the indicator functions  $\mathbf{1}_{\lambda_i}$  can be replaced by triangles function such as in Figure 1. In fact, as long as a real map  $F_i$  matches with  $\mathbf{1}_{\lambda_i}$  on all the eigenvalues of  $\mathcal{S}$ , we have the equality  $\mathbf{1}_{\lambda_i}(\mathcal{S}) = F_i(\mathcal{S})$ . Since the spectrum of  $\mathcal{S}$  is discrete, functions  $F_i$  as defined in Figure 1 are valid candidates for that.

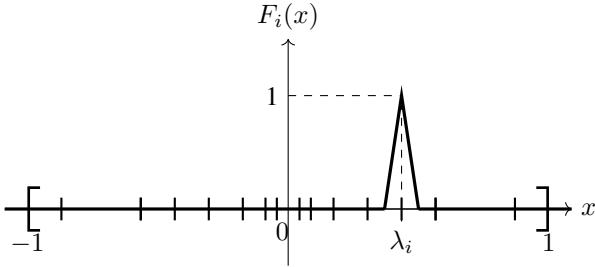


Fig. 1. The function  $F_i$  is defined as a triangle function whose angle at the peak is tight enough to match with  $\mathbf{1}_{\lambda_i}$  on the spectrum of  $\mathcal{S}$ .

These functions  $F_i$  can be easily parametrised using two parameters: the position and the angle of the peak. Moreover, they do not rely on the knowledge of the eigenvalues  $\lambda_i$ . As a consequence, the following family is also universal on compacts  $\mathcal{K} \subset \mathcal{O}^+$ :

$$\left\{ f \mapsto \sum_{i=1}^n h^{\text{MLP}} \left( \left( \|F_i(\mathcal{S})f\|_{i=1}^n \right) \frac{F_i(\mathcal{S})f}{\|F_i(\mathcal{S})f\|} \right) \right\}.$$

**Remark 1.** Let  $F \in \mathcal{C}(\sigma(\mathcal{S}), \mathbf{R})$ , since

$$F(\mathcal{S}): f \mapsto \sum_{i=1}^{\infty} F(\lambda_i) \langle f, \varphi_i \rangle \varphi_i,$$

we have that for all  $f \in \mathcal{O}^+$ ,  $F(\mathcal{S})(f) = 0$  if and only if  $F(\lambda_i) = 0$  for all  $i$ , that is,  $F(\mathcal{S})$  is the identically null operator. Therefore, in the sequel, we adopt the notational convention that, for  $f \in \mathcal{O}^+$ ,

$$\frac{F(\mathcal{S})(f)}{\|F(\mathcal{S})(f)\|} = 0$$

if  $F(\mathcal{S}) = 0_{\mathcal{L}(\mathcal{H})}$ .

2) *Polynomial filter approximation*: In practice, polynomial filters are used for convolutional neural networks. In this section, we replace the indicators from Equation (10) by polynomial approximations, as a consequence of Stone-Weierstrass theorem. Our result rely on the following lemmas.

**Lemma 1.** Let  $i \geq 0$ , for any compact  $\mathcal{K} \subset \mathcal{H}$ , for any  $\varepsilon > 0$ , there is a polynomial  $Q$  such that

$$\sup_{f \in \mathcal{K}} \|Q(\mathcal{S})(f) - \mathbf{1}_{\lambda_i}(\mathcal{S})(f)\| \leq \varepsilon. \quad (13)$$

**Lemma 2.** Let  $i \geq 0$ , for any compact  $\mathcal{K} \subset \mathcal{O}^+$ , for any  $\varepsilon > 0$ , there is a polynomial  $R$  such that

$$\sup_{f \in \mathcal{K}} \left\| \frac{R(\mathcal{S})(f)}{\|R(\mathcal{S})(f)\|} - \varphi_i \right\| \leq \varepsilon. \quad (14)$$

From these lemmas, we obtain that the following family is universal for  $\mathcal{K} \subset \mathcal{O}^+$ :

$$\left\{ f \mapsto \sum_{i=1}^n h^{\text{MLP}} \left( \left( \|Q_i(\mathcal{S})f\|_{i=1}^n \right) \frac{R_i(\mathcal{S})f}{\|R_i(\mathcal{S})f\|} \right) \right\}. \quad (15)$$

Finally, remark that this last family can be considered as a family of cGNNs (3): they can perform polynomial filtering by several rounds of message-passing, and all operations required to compute continuous transformations such as the inner product, the  $L^2$  norm, multiplication, square root, and so on, can be approximated by MLPs and global pooling to integrate. At the end of the day, we obtain the following theorem.

**Theorem 1.** Assume that  $\mathcal{S}$  is compact, full-rank, with simple eigenvalues. Then, cGNNs are universal for  $\mathcal{K} \subset \mathcal{O}^+$ .

## V. CONCLUSION AND DISCUSSION

In this work, we proposed a notion of universality for cGNNs based on the capability to approximate non-linear continuous operators between function spaces. Since the limits of GNNs on large random graphs are instances of such operators, in their equivariant form, where both the input and the output spaces are function spaces, we investigated the possibility to build universal cGNN families. Eventually, we obtained a family of this kind in Theorem 1. Yet, we acknowledge that our result necessitate strong assumptions as of now, and opens many paths for future investigations. Let us describe some of them.

In Equation (8), we faced the issue of sign ambiguity of the eigenvectors as well possible null components on some eigenspaces. We circumvented this difficulty by defining the set  $\mathcal{O}^+$  and deciding to restrict to compact subsets of  $L^2(\mathcal{X}, P)$  contained in  $\mathcal{O}^+$ . This is quite a strong assumption. A possible solution to circumvent this could be to use what is known as *positional encoding*. Positional encoding is a method that aims to enhance the expressive power by preprocessing. The idea is that instead of feeding a model with raw data, we feed a version of this data that has been judiciously transformed, in the hope that it would help the model in its decision process. In the context of graph learning a popular positional encoding is to make use of the eigenvectors of the Laplacian (or another

suitable representative operator at hand). In practice, one has to be careful to the bias induced when making an arbitrary choice for the eigenbasis of the Laplacian. In the case of single-dimensional eigenspace, as we have assumed here, this bias is precisely a sign ambiguity. The authors in Lim *et al.* [19] proposed to alleviate this issue with an architecture that is invariant to sign flip of eigenvector. See also Keriven and Vaiter [10] for a highlight of positional encoding influence in terms of expressiveness. For example, we could try to replace the eigenfunctions from Equation (8) by a sign invariant version

$$\varphi_i \leftarrow h_i^{\text{MLP}} \circ \varphi_i + h_i^{\text{MLP}} \circ (-\varphi_i)$$

where,  $h_i^{\text{MLP}}$  are arbitrary MLPs. The problem will be to determine the approximation power of such operators.

Another unexplored path relates to the use of other *message passing* operations, such as graph attention [20] (GAT), which are known to produce converging GNNs on random graphs [7]. We restricted to graph filtering in this study, where  $\mathcal{S}$  is a linear operator. On the contrary, continuous GAT generalizes to non-linear operators [7], which may solve some of the issues we encountered in our studies.

#### ACKNOWLEDGEMENT

This work was partially supported by the French National Research Agency in the framework of the « France 2030 » program (ANR-15-IDEX-0002), the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01), and the ANR grants GRANDMA (ANR-21-CE23-0006), GRANOLA (ANR-21-CE48-0009), GraVa (ANR-18-CE40-0005), and MAD (ANR-24-CE23-1529).

#### REFERENCES

- [1] B. Weisfeiler and A. Leman, “The reduction of a graph to canonical form and the algebra which appears therein,” *nti Series*, vol. 2, no. 9, pp. 12–16, 1968.
- [2] R. Levie, E. Isufi, and G. Kutyniok, “On the transferability of spectral graph filters,” in *2019 13th International conference on Sampling Theory and Applications (SampTA)*, IEEE, 2019, pp. 1–5.
- [3] N. Keriven, A. Bietti, and S. Vaiter, “Convergence and Stability of Graph Convolutional Networks on Large Random Graphs,” in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 21 512–21 523.
- [4] L. Ruiz, L. Chamon, and A. Ribeiro, “Graphon neural networks and the transferability of graph neural networks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1702–1712, 2020.
- [5] P. D. Hoff, A. E. Raftery, and M. S. Handcock, “Latent space approaches to social network analysis,” *Journal of the American Statistical Association*, vol. 97, no. 460, pp. 1090–1098, 2002.
- [6] L. Ruiz, F. Gama, and A. Ribeiro, “Graph neural networks: Architectures, stability, and transferability,” *Proceedings of the IEEE*, vol. 109, no. 5, pp. 660–682, 2021. DOI: 10.1109/JPROC.2021.3055400.
- [7] M. Cordonnier, N. Keriven, N. Tremblay, and S. Vaiter, “Convergence of message passing graph neural networks with generic aggregation on large random graphs,” *Journal of Machine Learning Research*, 2024. arXiv: 2304.11140 [stat.ML].

- [8] S. Maskey, R. Levie, Y. Lee, and G. Kutyniok, “Generalization analysis of message passing neural networks on large random graphs,” in *Advances in Neural Information Processing Systems*, vol. 35, Curran Associates, Inc., 2022, pp. 4805–4817.
- [9] N. Keriven, A. Bietti, and S. Vaiter, “On the universality of graph neural networks on large random graphs,” in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 6960–6971.
- [10] N. Keriven and S. Vaiter, “What functions can graph neural networks compute on random graphs? the role of positional encoding,” in *Advances in Neural Information Processing Systems*, vol. 36, Curran Associates, Inc., 2023, pp. 11 823–11 849.
- [11] J. Böker, R. Levie, N. Huang, S. Villar, and C. Morris, “Fine-grained expressivity of graph neural networks,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [12] M. Cordonnier, “Contribution to the theory of graph neural networks on large random graphs,” *PhD Thesis*, 2024.
- [13] P. W. Holland, K. B. Laskey, and S. Leinhardt, “Stochastic blockmodels: First steps,” *Social Networks*, vol. 5, no. 2, pp. 109–137, 1983, ISSN: 0378-8733. DOI: [https://doi.org/10.1016/0378-8733\(83\)90021-7](https://doi.org/10.1016/0378-8733(83)90021-7).
- [14] P. Erdős, “Graph theory and probability,” *Canadian Journal of Mathematics*, vol. 11, pp. 34–38, 1959. DOI: 10.4153/CJM-1959-003-9.
- [15] M. Penrose, *Random Geometric Graphs*. Oxford University Press, May 2003, ISBN: 9780198506263. DOI: 10.1093/acprof:oso/9780198506263.001.0001.
- [16] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, PMLR, Jun. 2017, pp. 1263–1272.
- [17] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in *Proceedings of the 5th International Conference on Learning Representations*, (ICLR), ser. ICLR ’17, 2017.
- [18] N. Kovachki *et al.*, “Neural operator: Learning maps between function spaces with applications to pdes,” *J. Mach. Learn. Res.*, 2023.
- [19] D. Lim *et al.*, “Sign and basis invariant networks for spectral graph representation learning,” in *The Eleventh International Conference on Learning Representations*, 2023.
- [20] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” *6th International Conference on Learning Representations*, 2017.