

Convolutional Filters and Neural Networks With Noncommutative Algebras

Alejandro Parada-Mayorga , Landon Butler , and Alejandro Ribeiro 

Abstract—In this paper we introduce and study the algebraic generalization of non commutative convolutional neural networks. We leverage the theory of algebraic signal processing to model convolutional non commutative architectures, and we derive concrete stability bounds that extend those obtained in the literature for commutative convolutional neural networks. We show that non commutative convolutional architectures can be stable to deformations on the space of operators. We develop the spectral representation of non commutative signal models to show that non commutative filters process Fourier components independently of each other. In particular we prove that although the spectral decompositions of signals in non commutative models are associated to eigenspaces of dimension larger than one, there exists a trade-off between stability and selectivity, which is controlled by matrix polynomial functions in spaces of matrices of low dimension. This tradeoff shows how when the filters in the algebra are restricted to be stable, there is a loss in discriminability that is compensated in the network by the pointwise nonlinearities. The results derived in this paper have direct applications and implications in non commutative convolutional architectures such as group neural networks, multigraph neural networks, and quaternion neural networks, for which we provide a set of numerical experiments showing their behavior when perturbations are present.

Index Terms—Non commutative convolutional architectures, algebraic neural networks (AlgNNs), algebraic signal processing (ASP), representation theory of algebras, non commutative algebras, non commutative operators, non commutative neural networks, Fréchet differentiability.

I. INTRODUCTION

DEEP learning relies on parameterizations given by the composition of layers which are themselves compositions of linear operators with pointwise nonlinearities. In problems that involve high dimensional inputs it becomes necessary to exploit their structure to reduce the complexity of the learning parametrization. This is often accomplished with the use of particular instantiations of *convolutional* filter banks. The most notable examples of this approach are the use of standard

(Euclidean) *convolutional* filters for learning with time signals and images, graph *convolutional* filters for learning with graphs and graph signals, and group *convolutions* for signals with group symmetries. The success of convolutional neural networks of different types – Euclidean, graph, and group – is in part due to how convolutions leverage symmetries of the domain and the signals. However, both filter banks and convolutional neural networks are equally good at leveraging these symmetries [1], [2], [3]. Yet, it is the latter that are more successful at learning. This empirical observation prompts a search for properties that explain the better performance of convolutional neural networks relative to the corresponding convolutional filter banks.

In the case of Euclidean convolutions and graph convolutions part of the insight into the relative performance of convolutional filter banks and neural networks follows from their respective responses to deformations [4], [2] – deformations in the domain of the signals in [4] and deformations on the space of graph matrix representations in [2]. These works prove that stability to deformations requires filters that do not distinguish different high frequency components. It follows that convolutional neural networks can better trade-off discriminability and stability as the pointwise nonlinearity mixes frequency components across layers. Recent analysis conducted on the operator space has demonstrated that the shared stability properties of Euclidean and graph convolutions are a consequence of their shared algebraic structure [1]. It then follows that whenever we use convolutional parameterizations in machine learning we can expect convolutional neural networks with multiple layers to outperform convolutional filter banks. These results build on the theory of algebraic signal processing (ASP) which provides a common language for describing convolutional filters of different types [5]. They apply to Euclidean, graph, and group convolutions as well as to a large class of less ubiquitous convolutional filters and neural networks [1], [6].

Although these results provide valuable insights, they are limited to commutative operators. Namely, to signal processing architectures in which filters commute. This is not true in general as non commutative signal processing arises naturally in multiple scenarios [7], [8], [9], [10], [11], [12], [13], [14], [15]. For instance, multigraphs arise when nodes are related by several different types of edges [7], [16]. In this case it is natural to define multigraph filters by combining diffusions across matrix representations of the graphs defined by each individual type (see Example 3 in the supplementary material). The resulting filters do not commute except in the rare eventuality that the

Manuscript received 7 July 2022; revised 15 February 2023 and 17 June 2023; accepted 5 July 2023. Date of publication 14 July 2023; date of current version 16 August 2023. This work was supported by NSF-Simons MoDL Award #2031985. The associate editor coordinating the review of this manuscript and approving it for publication was Ivan Dokmanić. (Corresponding author: Alejandro Parada-Mayorga.)

Alejandro Parada-Mayorga, Landon Butler, and Alejandro Ribeiro are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: alejopm@seas.upenn.edu; landonb3@seas.upenn.edu; aribeiro@seas.upenn.edu).

This article has supplementary downloadable material available at <https://doi.org/10.1109/3293716>, provided by the authors.

Digital Object Identifier 10.1109/TSP.2023.3293716

different matrix representations themselves commute. Filters that do not commute also arise with signal models on groups like $\mathbb{E}(n)$, or $SO(n)$ [8], [9], [10], [17], or any nontrivial Lie group [11], [12], [17], [18]. Non commutative filters appear here because filters are associated with group symmetries and it is common for groups to have symmetries that do not commute – such as rotations in $SO(3)$.

Existing stability results do not apply to any of these non commutative settings but empirical results show the usual advantage of layered neural networks with respect to filter banks [11], [12], [15], [16], [17]. Such an empirical observation poses the question of whether similar stability results hold for non commutative convolutional architectures. That is, whether convolutional versions of architectures such as multigraph neural networks [16], [19], neural networks on groups [15], [17], quaternion (graph) neural networks [14], [20], [21], [22], [23], [24], [25], hypercomplex algebras neural networks [26], hyperbolic neural networks [27], [28], octonion neural networks [29], and Clifford algebras neural networks [30], [31], [32] can be stable to deformations and, more to the point, whether stability requires spectral restrictions analogous to those of Euclidean, graph, and commutative group neural networks. The goal of this paper is to answer this question in the affirmative. To do so our first two contributions are the following:

- (C1) We define convolutional filters (Section II) and neural networks (Section IV) with non commutative algebras.
- (C2) We develop the spectral representation of non commutative signal models (Section III).

Non commutative filters are just a particular case of algebraic filters [5] and the construction of algebraic neural networks with non commutative filters is a close to verbatim extension of algebraic neural networks [1], [6]. The development of spectral representations, however, has substantial differences. Commutative signal models define Fourier decompositions as projections on single dimensional subspaces and, consequently, frequencies are defined as scalars. In non commutative signal models Fourier decompositions are projections in multidimensional subspaces and frequencies are, consequently, matrices of corresponding dimensions (Definition 5). Despite this significant difference we can still prove the equivalent of an spectral representation theorem of non commutative algebraic filters:

- (C3) Non commutative algebraic filters process Fourier components independently of each other (Theorem 1).

Contribution (C3) is the equivalent of the claim that commutative algebraic filters process frequency components independently of each other. As in the case of commutative filters, Contribution (C3) implies that non commutative algebraic filters are completely characterized by their frequency responses. The difference is that in non commutative filters the frequency response is a *matrix* polynomial (Definition 6). This is in contrast to the scalar polynomials that define frequency representations in commutative signal models and a consequence of the fact that frequencies are matrices, not scalars.

It is this difference between frequency representations of commutative and non commutative filters – matrix versus scalar polynomials – that prevents the results derived in [1] to be applied to non commutative convolutional architectures. The main technical contribution of this paper is to generalize the analysis

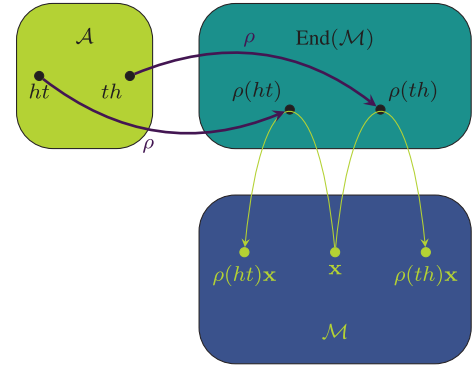


Fig. 1. Non commutative algebraic signal model. The algebraic filters th and ht are realized physically in $\text{End}(\mathcal{M})$ to process the signals $\mathbf{x}$ which are modeled as elements of $\mathcal{M}$.

of [1] to show that analogous results hold for non commutative algebraic signal models. In particular, we prove that:

- (C4) Non commutative algebraic convolutional filters *can* be stable to additive and multiplicative perturbations of the algebraic signal model (Section VI).
- (C5) Non commutative algebraic neural networks inherit the stability properties of algebraic convolutional filters (Section VI-A).

The proof of Contribution (C4) and (C5) requires that filters have two spectral properties (Definition 9). The first property is a generalized Lipschitz condition in which changes in the filter's frequency response are upper bounded by a linear function of the frequency's norm. The second condition is that the Fréchet derivative of the filter's response acting on a frequency matrix has bounded norm. These two conditions limit the discriminability of non commutative filters. They respectively imply that: (i) The variability of a stable filter must be bounded by the norm of the difference between frequency matrices. (ii) The variability of stable filters must decrease with the norm of the frequency matrix. Stability of commutative filters requires analogous conditions on scalar frequencies. Since imposing conditions on frequency matrices is more stringent, our stability results suggest that non commutative filters require more stringent filter restrictions to attain the same level of stability.

To illustrate our theoretical results we provide a set of numerical experiments for non commutative convolutional multigraph and quaternion neural networks (Section VII).

II. NON COMMUTATIVE FILTERS

In this section we introduce the generalization of non commutative signal models on arbitrary domains under the lens of algebraic signal processing (ASP). An algebraic signal model (ASM) is defined as the triplet

$$(\mathcal{A}, \mathcal{M}, \rho), \quad (1)$$

where $\mathcal{A}$ is a unital associative algebra, $\mathcal{M}$ is a vector space, and $\rho : \mathcal{A} \rightarrow \text{End}(\mathcal{M})$ is a homomorphism between the algebra $\mathcal{A}$ and the set of endomorphisms in the vector space $\mathcal{M}$ [5]. We recall that an algebra is a vector space with a closed operation of product, the endomorphisms of $\mathcal{M}$ are the set of linear maps from $\mathcal{M}$ onto itself and a homomorphism is a linear

map between algebras that preserves the product operation. We refer the reader to Appendix A for a review of these concepts and the discussion of some examples as well.

For our discussion we consider that $\mathcal{A}$ and $\mathcal{M}$ as vector spaces are defined on an *algebraically closed field* $\mathbb{F}$ which for the sake of simplicity will be considered $\mathbb{F} = \mathbb{C}$. However, we remark that the material discussed is valid for any algebraically closed Field. In the triplet $(\mathcal{A}, \mathcal{M}, \rho)$ the pair $(\mathcal{M}, \rho)$ constitutes a *representation* of $\mathcal{A}$ in the context of *representation theory of algebras*.

The processing of information in the context of (1) is known as algebraic signal processing (ASP). The signals are modeled as elements of $\mathcal{M}$, the filters are defined as elements of $\mathcal{A}$ and their realization is given by ρ . The rules or laws governing the operations are given by the structure of $\mathcal{A}$ and ρ translates those operations into actions on the elements of $\mathcal{M}$. Then, the filtering of a signal $\mathbf{x} \in \mathcal{M}$ by an algebraic filter $a \in \mathcal{A}$ is given by

$$\mathbf{y} = \rho(a)\mathbf{x}. \quad (2)$$

In (2) we have a generalized representation of the convolution operation between a filter and a signal. Particular instantiations of (2) lead to the traditional signal processing models of time signals and images and more sophisticated models such as graph signal processing (GSP), graphon signal processing (WSP) among others [1], [6], [33].

Since it is $\mathcal{A}$ the algebraic object defining the structural rules of information processing, the non commutativity in an algebraic model is associated to $\mathcal{A}$. This is, any non commutative convolutional signal model is given by $(\mathcal{A}, \mathcal{M}, \rho)$ where $\mathcal{A}$ is a non commutative algebra. For the reader unfamiliar with algebraic concepts, we remark that one representative example of this type of algebra is $M_{n \times n}(\mathbb{C})$, which is the set of matrices of dimension $n \times n$ with entries in $\mathbb{C}$. It is an algebra, since it is a vector space over $\mathbb{C}$ and the product of two matrices of size $n \times n$ is again a matrix of the same size.

Now we recall the notion of generators in algebraic signal models.

Definition 1: (Generators). For an associative algebra with unity $\mathcal{A}$ we say the set $\mathcal{G} \subseteq \mathcal{A}$ generates $\mathcal{A}$ if all $a \in \mathcal{A}$ can be represented as polynomial functions of the elements of $\mathcal{G}$. We say elements $g \in \mathcal{G}$ are generators of $\mathcal{A}$ and we denote as $a = p(\mathcal{G})$ the polynomial that generates a .

Representing the elements of $\mathcal{A}$ as polynomial functions of a set of generators highlights the fact that such generators characterize the algebra, and can be conceived as a measure of the degrees of freedom associated to the algebra. Consequently, the realization of the generators by means of the homomorphism ρ , called shift operators, play a central role when analyzing algebraic signal models. We introduce its formal definition next.

Definition 2: (Shift Operators). Let $(\mathcal{A}, \mathcal{M}, \rho)$ be an algebraic signal model. Then, if $\mathcal{G} \subset \mathcal{A}$ is a generator set of $\mathcal{A}$, the operators $\mathbf{S} = \rho(g)$ with $g \in \mathcal{G}$ are called shift operators. The set of all shift operators is denoted by $\mathcal{S}$.

Then, we can express any element $\rho(a) \in \text{End}(\mathcal{M})$ as a polynomial function of the shift operators. In particular, if $a = p(\mathcal{G})$ we have that

$$\rho(a) = p_{\mathcal{M}}(\rho(\mathcal{G})) = p_{\mathcal{M}}(\mathcal{S}) = p(\mathcal{S}), \quad (3)$$

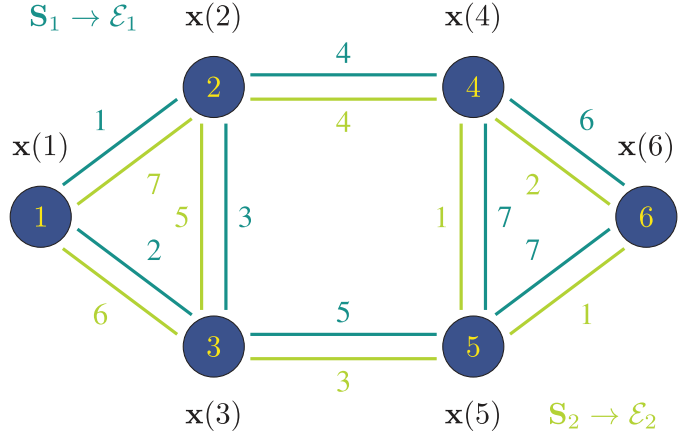


Fig. 2. Example of a multigraph signal processing model. The signal $\mathbf{x} \in \mathbb{R}^6$ defined on the multigraph $G = (\mathcal{V}, \{\mathcal{E}_1, \mathcal{E}_2\})$ is associated to the nodes $\mathcal{V}$, while the shift operator $\mathbf{S}_i$ is associated to the set of edges $\mathcal{E}_i$.

where $p_{\mathcal{M}}$ indicates the polynomial of form p but whose independent variables are the shift operators. For the sake of simplicity we drop the subindex in p in our subsequent discussion since it is clear where p is defined from the independent variable.

To facilitate the understanding of these basic concepts we present an example.

Example 1: (Multigraph signal processing). A multigraph consists of a common set of vertices and multiple separate sets of edges. Consider the specific case $G = (\mathcal{V}, \{\mathcal{E}_1, \mathcal{E}_2\})$ consisting of a set of N vertices $\mathcal{V}$ and two separate edge sets $\mathcal{E}_1$ and $\mathcal{E}_2$ – see Fig. 2. Associated with each edge set we consider matrix representations $\mathbf{S}_1$ and $\mathbf{S}_2$. This is an algebraic signal processing model in which the vector space of signals $\mathcal{M} = \mathbb{C}^N$ contains vectors in $\mathbb{C}^N$ with entries associated with each node of the multigraph and the algebra $\mathcal{A} = \mathbb{C}[t_1, t_2]$ is the set of non commutative polynomials of two variables. The homomorphism ρ is defined by mapping the generators t_1 and t_2 to $\rho(t_1) = \mathbf{S}_1$ and $\rho(t_2) = \mathbf{S}_2$, where the shift operators $\mathbf{S}_i$ are the matrix representations of the corresponding set of edges $\mathcal{E}_i$. Then, if $p(t_1, t_2) = t_1^2 + t_1 t_2 + 2t_2 t_1 + t_2^2 + 1$ the filtering in (2) takes the form

$$\begin{aligned} & \rho(t_1^2 + t_1 t_2 + 2t_2 t_1 + t_2^2 + 1) \mathbf{x} \\ &= (\mathbf{S}_1^2 + \mathbf{S}_1 \mathbf{S}_2 + 2\mathbf{S}_2 \mathbf{S}_1 + \mathbf{S}_2^2 + 1) \mathbf{x}. \end{aligned} \quad (4)$$

We refer the reader to Appendix F in the supplementary material for a more sophisticated example on multigraph signal processing in the light of ASP. In Appendix B and Appendix C of the paper it is shown how classical group signal processing and convolutional quaternion signal processing can be seen as particular cases of algebraic signal models.

Remark 1: It is important to highlight that the definition of an algebraic signal model is independent of any norm that could be associated to $\mathcal{M}$, if any. As we will discuss in subsequent sections (see Section VI) we will endow $\mathcal{M}$ with a norm in order to analyze the size of perturbations, and doing this will not affect the structure of the triplet $(\mathcal{A}, \mathcal{M}, \rho)$. We also remark that the algebra $\mathcal{A}$ is not associated to a norm, but it happens that $\mathcal{A}$

can be isomorphic to algebras that are naturally endowed with a norm. This is something we exploit in Section III to characterize subsets of the algebra and spectral representations.

III. SPECTRAL REPRESENTATIONS OF NON COMMUTATIVE FILTERS

In this section we discuss spectral representations of filters and signals. We will show that spectral representations in non commutative signal models are determined by matrix polynomials and that frequencies themselves are described by matrices instead of scalars. This introduces technical challenges, and yet we will also show that despite the significant differences with respect to the commutative scenario, spectral filtering is an analog of spectral filtering in commutative signal models. Additionally, this analog behavior extends to the norms of the operators, in particular the norm of non commutative filters is determined by the norm of the spectral responses which can be characterized in analogous ways to those in commutative signal models when considering induced norm on direct sums of spaces.

The notion of spectral decomposition descends from the concepts of irreducible and completely reducible subrepresentations of an algebra [5], [34], [35], [36], [37]. Widely known notions of spectral decompositions used in graph signal processing (GSP), discrete signal processing (DSP), and discrete time signal processing (DTSP) among others, are obtained as particular cases of a decomposition of representations of an algebra as a sum of irreducible subrepresentations. Then, in order to present a natural generalization of frequency decompositions in non commutative ASM, we discuss the concepts of subrepresentation, irreducibility and decomposability. We restrict our attention to algebras that have a finite number of generators.

Definition 3: Let $(\mathcal{M}, \rho)$ be a representation of $\mathcal{A}$. Then, a representation $(\mathcal{U}, \rho)$ of $\mathcal{A}$ is a subrepresentation of $(\mathcal{M}, \rho)$ if $\mathcal{U} \subseteq \mathcal{M}$ and $\mathcal{U}$ is invariant under all operators $\rho(a)$ for all $a \in \mathcal{A}$, i.e. $\rho(a)u \in \mathcal{U}$ for all $u \in \mathcal{U}$ and $a \in \mathcal{A}$.

The notion of subrepresentation is tied to the property of invariance since subrepresentations are invariant under the action of the instantiations of elements of $\mathcal{A}$ in $\text{End}(\mathcal{M})$. Notice that when considering representations of an algebra with a single generator, the spaces generated by subsets of eigenvectors of $\rho(a)$ determine the subrepresentations of $(\mathcal{M}, \rho)$.

Now, we introduce formally irreducible subrepresentations which provide a minimal structural unit of invariance.

Definition 4: A representation $(\mathcal{M}, \rho)$ (with $\mathcal{M} \neq 0$) is irreducible or simple if the only subrepresentations of $(\mathcal{M}, \rho)$ are $(0, \rho)$ and $(\mathcal{M}, \rho)$.

The irreducibility property of a subrepresentation implies that there is not a subspace that has its own invariance under the action of the elements of the algebra. Additionally, notice that one dimensional subrepresentations are always irreducible. It is important to point out that the calculation of irreducible subrepresentations entails a significant computational cost – see Appendix K in the supplementary material.

As we will show in the next subsection, irreducibility can be used to write a general representation as a *decomposition*

in terms of irreducible subrepresentations. To show this we introduce the notion of direct sum of representations. Given two representations $(\mathcal{M}_1, \rho_1)$ and $(\mathcal{M}_2, \rho_2)$ of an algebra $\mathcal{A}$ we can obtain a new representation, called the *direct sum representation*, as $(\mathcal{M}_1 \oplus \mathcal{M}_2, \rho)$ where $\rho(a)(u_1 \oplus u_2) = \rho_1(a)u_1 \oplus \rho_2(a)u_2$.

A. Fourier Decompositions and Spectral Representation of Filters

Using the ideas and concepts discussed above we introduce the notion of Fourier decomposition in algebraic signal processing.

Definition 5: (Fourier Decomposition). For an algebraic signal model $(\mathcal{A}, \mathcal{M}, \rho)$ we say that there is a spectral or Fourier decomposition if

$$(\mathcal{M}, \rho) \cong \bigoplus_{(\mathcal{U}_i, \phi_i) \in \text{Irr}\{\mathcal{A}\}} (\mathcal{U}_i, \phi_i), \quad (5)$$

where the $(\mathcal{U}_i, \phi_i)$ are irreducible subrepresentations of $(\mathcal{M}, \rho)$. Any signal $\mathbf{x} \in \mathcal{M}$ can be therefore represented by the map Δ given by

$$\begin{aligned} \Delta : \mathcal{M} &\rightarrow \bigoplus_{(\mathcal{U}_i, \phi_i) \in \text{Irr}\{\mathcal{A}\}} \mathcal{U}_i \\ \mathbf{x} &\mapsto \hat{\mathbf{x}}, \end{aligned} \quad (6)$$

known as the Fourier decomposition of $\mathbf{x}$ and the projection of $\hat{\mathbf{x}}$ in each $\mathcal{U}_i$ are the Fourier components represented by $\hat{\mathbf{x}}(i)$.

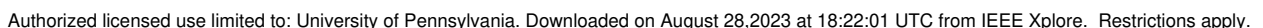
In Definition 5 it is assumed that each individual subrepresentation $(\mathcal{U}_i, \phi_i)$ cannot be expressed as a direct sum of irreducible subrepresentations isomorphic to $(\mathcal{U}_i, \phi_i)$. This assumption is analogous to the assumption of not having repeated eigenvalues when considering spectral decompositions where the irreducible subrepresentations have dimension equal to 1. This choice is done for the sake of simplicity and we refer the reader to Appendix K of the supplementary material for a more sophisticated formulation of the Fourier decompositions. We remark that the decomposition in (5) and (6) is unique up to isomorphism – see Appendix K in the supplementary material.

It is worth pointing out that the homomorphisms ϕ_i associated to each non isomorphic irreducible representation define the *frequency* associated to the vector space $\mathcal{U}_i$ [5] – see Fig. 3. In commutative scenarios, like for instance in GSP, we have $\dim(\mathcal{U}_i) = 1$. In this case the term $\phi_i(a)$ is a scalar value which corresponds to the eigenvalues of $\rho(a)$ while $\mathcal{U}_i$ is the space spanned by the i -th eigenvector.¹ If $\dim(\mathcal{U}_i) > 1$ then $\phi_i(a)$ is a matrix. This last scenario is typical when non commutative algebras are involved.

We synthesize the role of Fourier decompositions and the filtering operation in the following theorem.

Theorem 1: (Filtering Spectral Theorem) Let $(\mathcal{A}, \mathcal{M}, \rho)$ be an algebraic signal model where $\mathcal{A}$ has generators $\{g_k\}_{k=1}^m$, and

¹We refer the reader to Appendix G in the supplementary material where we show with more details how the Fourier transform in GSP can be obtained as a particular case of Definition 5.



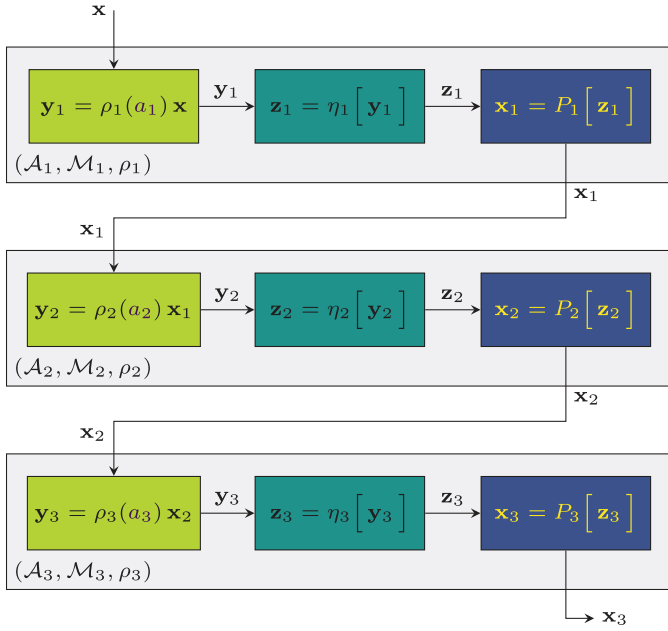


Fig. 5. Algebraic Neural Network $\{(\mathcal{A}_\ell, \mathcal{M}_\ell, \rho_\ell; \sigma_\ell)\}_{\ell=1}^3$ with three layers indicating how the input signal $\mathbf{x}$ is processed and mapped into $\mathbf{x}_3$. In each layer the information is transformed by a convolutional filter $\rho_\ell(a_\ell)$, followed by a pointwise nonlinearity η_ℓ and a pooling operator P_ℓ .

IV. ALGEBRAIC NEURAL NETWORKS WITH NON COMMUTATIVE ALGEBRAS

Algebraic neural networks (AlgNNs) are stacked layered structures (see Fig. 5) where the processing of information in each layer is carried out by means of filters of an algebraic signal model and pointwise nonlinear operators. For a training set $\mathcal{T} = \{\mathbf{x}, \mathbf{y}\}$ with inputs $\mathbf{x}$ and outputs $\mathbf{y}$, it is possible to learn the algebraic filters $a_\ell \in \mathcal{A}_\ell$ in each layer of the AlgNN to produce a mapping representation. This allows an estimation of the output to an unseen input $\tilde{\mathbf{x}} \in \mathcal{T}$. The data from the training set $\mathcal{T}$ is used to find subsets of filters $\mathcal{P}_\ell \subset \mathcal{A}_\ell$ that minimize a cost function of the form $\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{T}} f_{\mathcal{P}_\ell}(\mathbf{x}, \mathbf{y})$, where $f_{\mathcal{P}_\ell}(\mathbf{x}, \mathbf{y})$ is a fitting metric that penalizes the difference between $\mathbf{y}$ and the output of the AlgNN produced when the input is $\mathbf{x}$. In the ℓ -th layer of the AlgNN, an incoming signal $\mathbf{x}_{\ell-1}$ from the layer $\ell-1$ is filtered by means of the convolution $\rho_\ell(a_\ell)\mathbf{x}_{\ell-1}$. Then, a pointwise nonlinear operator $\eta_\ell : \mathcal{M}_\ell \rightarrow \mathcal{M}_\ell$ is applied and finally a pooling operator $P_\ell : \mathcal{M}_\ell \rightarrow \mathcal{M}_{\ell+1}$ matches information between $\mathcal{M}_\ell$ and $\mathcal{M}_{\ell+1}$. The output signal of the layer ℓ can be written as

$$\mathbf{x}_\ell = \sigma_\ell(\rho_\ell(a_\ell)\mathbf{x}_{\ell-1}) = \Phi(\mathbf{x}_{\ell-1}, \mathcal{P}_\ell, \mathcal{S}_\ell). \quad (10)$$

We use the symbol $\Phi(\mathbf{x}_{\ell-1}, \mathcal{P}_\ell, \mathcal{S}_\ell)$ to make emphasis in the fact that the filters used in each layer belong to specific subsets of the algebra and that a specific family of shift operators is being used. If several features per layer are used, we use the notation

$$\mathbf{x}_\ell^f = \sigma_\ell \left(\sum_{g=1}^{F_\ell} \rho_\ell(a_\ell^{gf}) \mathbf{x}_{\ell-1}^g \right), \quad (11)$$

where the super index f indicates the f th feature and ℓ the layer where information is being processed. To denote an AlgNN with L layers we use $\{(\mathcal{A}_\ell, \mathcal{M}_\ell, \rho_\ell; \sigma_\ell)\}_{\ell=1}^L$, where $\sigma_\ell = P_\ell \eta_\ell$. As pointed out in [1], traditional neural networks (CNNs), graph neural networks (GNNs), and graphon neural networks (WNNs) among others, can be obtained as particular instantiations of a general AlgNN with a commutative algebra. For our discussion we consider that $\sigma_\ell = P_\ell \eta_\ell$ is C_ℓ -Lipschitz and $\sigma_\ell(0) = 0$. The role of η_ℓ is crucial for the performance of any AlgNN, since it is the nonlinearity of η_ℓ what allows the AlgNN to redistribute spectral information and compensate restrictions imposed on the filters. There are many possible choices for η_ℓ that could improve the generalization capacity of the AlgNN. Typically a low computational cost η_ℓ is selected. The operator $P_\ell : \mathcal{M}_\ell \rightarrow \mathcal{M}_{\ell+1}$ performs an operation of dimensionality reduction. In particular instantiations of AlgNNs like GNNs, such operator can be associated to graph coarsening techniques or optimal sampling strategies for bandlimited signals [39]. In what follows we present some examples of concrete non commutative convolutional architectures instantiated as particular cases of non commutative algebraic neural networks (AlgNNs).

Remark 3: Note that the action of the pointwise nonlinearity, η_ℓ , and the pooling operator, P_ℓ , is defined in terms of a basis of $\mathcal{M}_\ell$. This is, given a basis $\{b_i^{(\ell)}\}_i$ of $\mathcal{M}_\ell$ we have $\eta_\ell \left(\sum_i \alpha_i b_i^{(\ell)} \right) = \sum_i \eta_\ell(\alpha_i) b_i^{(\ell)}$, and $P_\ell : \text{span}(\{b_i^{(\ell)}\}_i) \rightarrow \text{span}(\{b_i^{(\ell+1)}\}_i)$. In our analysis, the specifics of such basis are not relevant as long as the operator $\sigma_\ell = P_\ell \eta_\ell$ is Lipschitz and $\sigma_\ell(0) = 0$. Notice also that since P_ℓ performs dimensionality reduction, there are multiple choices for linear and nonlinear versions of P_ℓ depending on the properties of $\mathcal{M}_\ell$ [40], [41], [42] and we consider those choices for which $\sigma_\ell = P_\ell \eta_\ell$ is Lipschitz and $\sigma_\ell(0) = 0$. We point out however, that the optimal selection of such basis and the operators η_ℓ and P_ℓ opens up an interesting future research direction.

Example 2: (Multigraph neural networks). Let us consider a multigraph $G = (\mathcal{V}, \{\mathcal{E}_r\}_{r=1}^m)$ with set of nodes $\mathcal{V}$, $|\mathcal{V}| = N$ and multiset of edges $\{\mathcal{E}_r\}_{r=1}^m$. Let $\mathbf{S}_i$ be a matrix representation of the multigraph on the set of edges $\mathcal{E}_i$, which could be the adjacency matrix or a Laplacian matrix of the graph $(\mathcal{V}, \mathcal{E}_i)$. Then, the ℓ -th layer of a multigraph neural network is composed by convolutional operators followed by a pointwise nonlinearity and a pooling operator. The convolutional filters are given by the multivariable polynomials (see Example 3 in the supplementary material): $p(\mathbf{S}_1, \dots, \mathbf{S}_m)$. Then, given the input $\mathbf{x}_\ell$ to the ℓ -th layer of the multigraph neural network, we leverage symmetries on the multigraph to obtain $\mathbf{y}_\ell = p(\mathbf{S}_1, \dots, \mathbf{S}_m)\mathbf{x}_\ell$. After this, we apply a pointwise nonlinear operator $\eta_\ell : \mathbb{R}^N \rightarrow \mathbb{R}^N$ to obtain $\mathbf{z}_\ell = \eta_\ell(\mathbf{y}_\ell)$, with $\mathbf{z}_\ell(u) = \max\{\mathbf{y}_\ell(u), 0\}$, and where $\mathbf{z}_\ell(u)$ is the u -th component of $\mathbf{z}_\ell$. Besides the convolutional filter and the pointwise nonlinear operator, an operation of pooling may be considered to reduce the computational cost of processing with multiple filters. For multigraph neural networks, the pooling operator may be defined using the zeroing approach used for GNNs in [39], where the information is forced to be zero in a subset of nodes. This is done under the hypothesis that the

information in those nodes is less relevant or redundant. Then, if zeroing is used the pooling operator is given by $P_\ell : \mathbb{R}^N \rightarrow \mathbb{R}^N$ with $\mathbf{x}_{\ell+1} = P_\ell(\mathbf{z}_\ell)$ and $\mathbf{x}_{\ell+1}(u) = 0$ for any $u \in \mathcal{U} \subset \mathcal{V}$, and where the subset $\mathcal{U}$ could be chosen according to a specific heuristic or the minimization of a cost function. Notice that for the sake of simplicity, in this example we considered one filter per layer, but we can indeed use several filters.

We refer the reader to Appendix B and Appendix C for a discussion on how convolutional group neural networks and quaternion neural networks can be seen as particular cases of a generic AlgNN.

Remark 4: The cost of the convolution in the ℓ -th layer is $\mathcal{O}(N_\ell^2 m^K F_\ell G_\ell)$, where $N_\ell = \dim(\mathcal{M}_\ell)$ is the dimension of the vector space, m is the number of generators in the algebra, F_ℓ is the number of input features, G_ℓ is the number of output features, and K is the order of the polynomial filters. The number of learnable parameters in each layer is $\mathcal{O}(m^K F_\ell G_\ell)$, which does not depend on the dimension of the vector space. It is important to remark that the complexity associated with the learnable parameters can be reduced when adapting pruning algorithms like the one proposed in [16], [19], which reduces efficiently the number of monomials in an polynomial operator.

V. ALGEBRAIC PERTURBATION MODELS

For our discussion we consider perturbations of the generic algebraic signal model $(\mathcal{A}, \mathcal{M}, \rho)$ determined by a perturbation of ρ . As discussed in [1], if the realization of algebraic filters in the algebra is achieved by ρ , it is natural to consider that mismatches in the model occur on ρ . In the following definition we state formally the notion of perturbation in the context of ASP.

Definition 7: (ASP Model Perturbation [1]) Let $(\mathcal{A}, \mathcal{M}, \rho)$ be an ASP model with algebra elements generated by $g \in \mathcal{G}$ (Definition 1) and recall the definition of the shift operators $\mathbf{S} = \rho(g)$ (Definition 2). We say that $(\mathcal{A}, \mathcal{M}, \tilde{\rho})$ is a perturbed ASP model if for all $a = p(g)$ we have that

$$\tilde{\rho}(a) = p(\tilde{\rho}(g)) = p(\tilde{\mathbf{S}}), \quad (12)$$

where $\tilde{\mathbf{S}}$ is a set of perturbed shift operators of the form

$$\tilde{\mathbf{S}} = \mathbf{S} + \mathbf{T}(\mathbf{S}), \quad (13)$$

for all shift operators $\mathbf{S} \in \mathcal{S}$.

From Definition 7 we can see that the effect of a perturbation on the homomorphism ρ is expressed in terms of perturbations of the shift operators, which at the same time produce a perturbed version of any algebraic filter. Notice that $\tilde{\rho}$ is a general map that is not necessarily a homomorphism, but it could be. As shown in [1] this notion of perturbation is associated to practical scenarios in GSP, WSP and group signal processing.

The notion of stability is tied to the concept of size of a deformation. To measure the size of those deformations on the space of operators we use norms. In particular, we use the operator norms induced by a norm associated to $\mathcal{M}$. If $\dim(\mathcal{M}) < \infty$ the specifics of the norm in $\mathcal{M}$ are not relevant since all norms are equivalent [43]. If $\mathcal{M}$ is infinite dimensional we select a norm that guarantees that every bounded operator

in $\text{End}(\mathcal{M})$ is Hilbert-Schmidt and that the induced norm in a direct sum of replicas of $\mathcal{M}$ satisfies the maximum property – see Appendix H in the supplementary material. We remark that this does not affect the structure of any algebraic signal model $(\mathcal{A}, \mathcal{M}, \rho)$. We now state the formal definition of stability considered in our analysis.

Definition 8: (Operator Stability [1]) Given operators $p(\mathbf{S})$ and $p(\tilde{\mathbf{S}})$ defined on the models $(\mathcal{A}, \mathcal{M}, \rho)$ and $(\mathcal{A}, \mathcal{M}, \tilde{\rho})$ (cf. Definition 7) we say the operator $p(\mathbf{S})$ is Lipschitz stable if there exist constants $C_0, C_1 > 0$ such that

$$\begin{aligned} \|p(\mathbf{S})\mathbf{x} - p(\tilde{\mathbf{S}})\mathbf{x}\| &\leq \left[C_0 \sup_{\mathbf{S} \in \mathcal{S}} \|\mathbf{T}(\mathbf{S})\| \right. \\ &\quad \left. + C_1 \sup_{\mathbf{S} \in \mathcal{S}} \|D_{\mathbf{T}}(\mathbf{S})\| + \mathcal{O}(\|\mathbf{T}(\mathbf{S})\|^2) \right] \|\mathbf{x}\|, \end{aligned} \quad (14)$$

for all $\mathbf{x} \in \mathcal{M}$. In (14) $D_{\mathbf{T}}(\mathbf{S})$ is the Fréchet derivative of the perturbation operator $\mathbf{T}$.

The right hand side of (14) provides a measure of the deformation produced by $\mathbf{T}(\mathbf{S})$. Then, Definition 8 states that a given operator is stable to a perturbation $\mathbf{T}(\mathbf{S})$ if the deformation induced in the operator is proportional to the size of the deformation. For our discussion we consider the perturbation model given by

$$\mathbf{T}(\mathbf{S}_i) = \mathbf{T}_{0,i} + \mathbf{T}_{1,i}\mathbf{S}_i, \quad (15)$$

which is composed of an absolute or additive perturbation $\mathbf{T}_{i,0}$ and a relative or multiplicative perturbation $\mathbf{T}_{i,1}\mathbf{S}_i$. The family of perturbations is ruled by the condition

$$\|\mathbf{T}_{i,r}\|_F \leq \delta \|\mathbf{T}_{i,r}\|, \quad (16)$$

where $\delta > 0$. This is, the Frobenius norm of the perturbation operators is bounded by a scalar factor of the operator norm.

Remark 5: The notion of stability discussed here is stability to deformations on the *operator* space. This is the same kind of deformation that is studied in [1] for commutative algebraic filters and in [2] for graph filters. It is not the same as the *domain* deformations studied in [4]. Although different in principle, both notions can be related [1]. For instance, a small deformation of the time axis for time signals implies that signals will not have translation symmetry but instead quasi translation symmetry. This can be seen as a perturbation of the time delay operator. It is important to point out that in all scenarios – domain and operator deformations – a stable operators must have changes that are proportional in size to the given deformations. This is done by using metrics used to measure the size of the diffeomorphisms involved in each case [44], [45], [46].

VI. STABILITY THEOREMS

In the following definitions we state properties of algebraic filters in $\mathcal{A}$ in terms of their spectral representations in $\tilde{\mathcal{A}}$ (see Def. 6). This will be used in the derivation of the stability results. We start introducing the notions of Lipschitz and integral Lipschitz filters.

Definition 9: Let $(\mathcal{A}, \mathcal{M}, \rho)$ be an ASM where $\mathcal{A}$ has generators $\{g_i\}_{i=1}^m$. Let $\tilde{\mathcal{A}}$ the algebra of matrices containing the spectral representations of the elements in $\mathcal{A}$ (see Def. 6).

We say that $p(g_1, \dots, g_m) \in \mathcal{A}$ is L_0 -Lipschitz if there exists $L_0 > 0$ such that

$$\begin{aligned} & \|p(x_1, \dots, x_m) - p(\tilde{x}_1, \dots, \tilde{x}_m)\| \\ & \leq L_0 \|(x_1, \dots, x_m) - (\tilde{x}_1, \dots, \tilde{x}_m)\|, \end{aligned} \quad (17)$$

for all $x_i, \tilde{x}_i$ and where $p(x_1, \dots, x_m) \in \tilde{\mathcal{A}}$. Additionally, it is said that $p(g_1, \dots, g_m) \in \mathcal{A}$ is L_1 -integral Lipschitz if there exists $L_1 > 0$ such that

$$\|D_{p|x_i}(x_1, \dots, x_m) \{(\cdot) x_i\}\| \leq L_1, \quad (18)$$

for all x_i , where $D_{p|x_i}(x_1, \dots, x_m)$ is the partial Fréchet derivative of $p(x_1, \dots, x_m) \in \tilde{\mathcal{A}}$, and $\|\cdot\|$ is the operator norm.

From now on we denote the set of algebraic Lipschitz filters by $\mathcal{A}_{L_0}$ and the set of algebraic integral Lipschitz filters by $\mathcal{A}_{L_1}$.

Now we introduce the first stability theorem for algebraic signal models with non commutative algebras with multiple generators.

Theorem 2: Let $(\mathcal{A}, \mathcal{M}, \rho)$ be an ASM where $\mathcal{A}$ is generated by $\{g_i\}_{i=1}^m$ and let $\rho(g_i) = \mathbf{S}_i \in \text{End}(\mathcal{M})$ for all i . Let $\tilde{\rho}(g_i) = \tilde{\mathbf{S}}_i \in \text{End}(\mathcal{M})$, where $(\mathcal{A}, \mathcal{M}, \tilde{\rho})$ is a perturbed version of $(\mathcal{A}, \mathcal{M}, \rho)$ and $\tilde{\mathbf{S}}_i$ is related with $\mathbf{S}_i$ by the perturbation model in (13). Then, for any $p \in \mathcal{A}$ we have

$$\begin{aligned} & \|p(\mathbf{S}_1, \dots, \mathbf{S}_m)\mathbf{x} - p(\tilde{\mathbf{S}}_1, \dots, \tilde{\mathbf{S}}_m)\mathbf{x}\| \\ & \leq \|\mathbf{x}\| \sum_{i=1}^m (\|D_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m)\mathbf{T}(\mathbf{S}_i)\| + \mathcal{O}(\|\mathbf{T}(\mathbf{S}_i)\|^2)), \end{aligned} \quad (19)$$

where $D_{p|\mathbf{S}_i}(\mathbf{S})$ is the partial Fréchet derivative of p on $\mathbf{S}_i$.

Proof: See Appendix D-A. $\square$

It is worth pointing out that in (19) the upper bound adds the contributions of the deformation in each direction $\mathbf{S}_i$ given by the shift operators. Each individual contribution associated to the perturbation of $\mathbf{S}_i$ is determined by the Fréchet derivative of the filters acting on the perturbation, and this is true no matter what function $\mathbf{T}(\mathbf{S}_i)$ is being considered.

In the following theorems we provide the basic stability result for algebraic filters showing how with a restriction of the filters shaped by the functional form of the right hand side of (19) leads to stability.

Theorem 3: Let $(\mathcal{A}, \mathcal{M}, \rho)$ be an ASM where $\mathcal{A}$ is a non commutative algebra with m generators $\{g_i\}_{i=1}^m$. Let $(\mathcal{A}, \mathcal{M}, \tilde{\rho})$ a perturbed version of $(\mathcal{A}, \mathcal{M}, \rho)$ by means of the perturbation model in (15). Then, if $p \in \mathcal{A}_{L_0} \cap \mathcal{A}_{L_1}$ it holds that

$$\begin{aligned} & \|D_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m)\mathbf{T}(\mathbf{S}_i)\| \leq \delta L_0 \sup_{\mathbf{S}_i \in \mathcal{S}} \|\mathbf{T}(\mathbf{S}_i)\| \\ & + \delta L_1 \sup_{\mathbf{S}_i \in \mathcal{S}} \|D_{\mathbf{T}}(\mathbf{S}_i)\|. \end{aligned} \quad (20)$$

Proof: See Appendix D-B. $\square$

From Theorems 2 and 3 we can state the stability results for filters in algebraic models with multiple generators in the following corollary.

Corollary 1: Let $(\mathcal{A}, \mathcal{M}, \rho)$ be a non commutative ASM where $\mathcal{A}$ has generators $\{g_i\}_{i=1}^m$. Let $(\mathcal{A}, \mathcal{M}, \tilde{\rho})$ be a perturbed version of $(\mathcal{A}, \mathcal{M}, \rho)$ associated to the perturbation model in (15). If $p \in \mathcal{A}_{L_0} \cap \mathcal{A}_{L_1} \subset \mathcal{A}$, the operator $p(\mathbf{S}_1, \dots, \mathbf{S}_m)$

is stable in the sense of Definition 8 with $C_0 = m\delta L_0$ and $C_1 = m\delta L_1$.

Proof: Replacing (20) from Theorem 3 into (19) from Theorem 2 and organizing the terms. $\square$

Notice that the stability property of the algebraic filters comes at the expense of their selectivity. A restriction on the subsets of the algebra naturally limits the type of frequency representations. Additionally, it is worth pointing out that it is the functional form of $\mathbf{T}(\mathbf{S}_i)$ the one feature that dictates the types of restrictions imposed on the algebra, necessary to guarantee stability.

Remark 6: We remark that the non commutativity imposes multidimensional frequency representations, but even in this scenario we have a trade-off between stability and selectivity. We could therefore say that although the non commutativity of the algebra does not change the fact that filters can be stable and selective, the multidimensional nature of the frequency representations leads to restrictions that can be captured only by means of matrix algebras. This in itself implies that two filters with identical functional form – symbolic expression – do not have the same stability properties if one of them is associated to a commutative model and the other to a non commutative one.

A. Stability of Non Commutative Algebraic Neural Networks

In this section we extend the stability results obtained for algebraic filters to operators representing AlgNNs. Let $\{(\mathcal{A}_\ell, \mathcal{M}_\ell, \rho_\ell; \sigma_\ell)\}_{\ell=1}^L$ be an AlgNN with L layers, and whose perturbed version is represented as $\{(\mathcal{A}_\ell, \mathcal{M}_\ell, \tilde{\rho}_\ell; \sigma_\ell)\}_{\ell=1}^L$. We start stating formally how the algebraic operators are affected by the functions that map information between layers.

Theorem 4: Let $\{(\mathcal{A}_\ell, \mathcal{M}_\ell, \rho_\ell; \sigma_\ell)\}_{\ell=1}^L$ be an algebraic neural network and $\{(\mathcal{A}_\ell, \mathcal{M}_\ell, \tilde{\rho}_\ell; \sigma_\ell)\}_{\ell=1}^L$ its perturbed version by means of the perturbation model in (15). We consider one feature per layer and non commutative algebras $\mathcal{A}_\ell$ with m generators. If $\Phi(\mathbf{x}_{\ell-1}, \mathcal{P}_\ell, \mathcal{S}_\ell)$ and $\Phi(\mathbf{x}_{\ell-1}, \mathcal{P}_\ell, \tilde{\mathcal{S}}_\ell)$ represent the ℓ -th mapping operators of the AlgNN and its perturbed version, it follows that

$$\begin{aligned} & \|\Phi(\mathbf{x}_{\ell-1}, \mathcal{P}_\ell, \mathcal{S}_\ell) - \Phi(\mathbf{x}_{\ell-1}, \mathcal{P}_\ell, \tilde{\mathcal{S}}_\ell)\| \\ & \leq C_\ell \delta \|\mathbf{x}_{\ell-1}\| m \left(L_0^{(\ell)} \sup_{\mathbf{S}_{i,\ell}} \|\mathbf{T}^{(\ell)}(\mathbf{S}_{i,\ell})\| \right. \\ & \quad \left. + L_1^{(\ell)} \sup_{\mathbf{S}_{i,\ell}} \|D_{\mathbf{T}^{(\ell)}}(\mathbf{S}_{i,\ell})\| \right) \end{aligned} \quad (21)$$

where C_ℓ is the Lipschitz constant of σ_ℓ , and $\mathcal{P}_\ell = \mathcal{A}_{L_0} \cap \mathcal{A}_{L_1}$ represents the domain of ρ_ℓ . The index (ℓ) makes reference to quantities and constants associated to the layer ℓ .

Proof: See Appendix J-A1 in the supplementary material. $\square$

Theorem 4 highlights the role of the maps σ_ℓ in the stability of the algebraic operators. In particular, we can see that the effect of such functions is only to scale the stability bound of the algebraic operators. If $C_\ell = 1$, the stability bounds for the operators in the layers of the AlgNN are identical to the stability bounds for the algebraic filters. However, it is important to

remark that the discriminability of the layer operators in the AlgNN is enriched by the pointwise nonlinearities.

Notice that the resemblance between the results provided in Theorem 4 and previous results for commutative architectures is rooted in the fact that the same notion of stability and the same type of perturbations are considered in both scenarios. However, there is a substantial difference between the stability bounds. In the non commutative scenario we have that the restrictions on the subsets of the algebra are dictated by matrix functions and not scalar maps. Additionally, as can be seen in the proof of the theorems above, the estimate of the upper bounds on the change of the operators requires a more general representation of spectral responses not given by the traditional spectral theorem.

Now we are ready to state the stability theorem for a general AlgNN.

Theorem 5: Let $\{(\mathcal{A}_\ell, \mathcal{M}_\ell, \rho_\ell; \sigma_\ell)\}_{\ell=1}^L$ be an algebraic neural network and $\{(\mathcal{A}_\ell, \mathcal{M}_\ell, \tilde{\rho}_\ell; \sigma_\ell)\}_{\ell=1}^L$ its perturbed version by means of the perturbation model in (15). We consider one feature per layer and non commutative algebras $\mathcal{A}_\ell$ with m generators. If $\Phi(\mathbf{x}, \{\mathcal{P}_\ell\}_1^L, \{\mathcal{S}_\ell\}_1^L)$ and $\Phi(\mathbf{x}, \{\mathcal{P}_\ell\}_1^L, \{\tilde{\mathcal{S}}_\ell\}_1^L)$ represent the mapping operator and its perturbed version, it follows that

$$\begin{aligned} & \left\| \Phi(\mathbf{x}, \{\mathcal{P}_\ell\}_1^L, \{\mathcal{S}_\ell\}_1^L) - \Phi(\mathbf{x}, \{\mathcal{P}_\ell\}_1^L, \{\tilde{\mathcal{S}}_\ell\}_1^L) \right\| \\ & \leq \sum_{\ell=1}^L \Delta_\ell \left(\prod_{r=\ell}^L C_r \right) \left(\prod_{r=\ell+1}^L B_r \right) \left(\prod_{r=1}^{\ell-1} C_r B_r \right) \|\mathbf{x}\|, \end{aligned} \quad (22)$$

where C_ℓ is the Lipschitz constant of σ_ℓ , $\|\rho_\ell(a)\| \leq B_\ell$ for all $a \in \mathcal{P}_\ell$, and $\mathcal{P}_\ell = \mathcal{A}_{L_0} \cap \mathcal{A}_{L_1}$ represents the domain of ρ_ℓ . The functions Δ_ℓ are given by

$$\Delta_\ell = \delta m \left(L_0^{(\ell)} \sup_{\mathbf{S}_{i,\ell}} \|\mathbf{T}^{(\ell)}(\mathbf{S}_{i,\ell})\| + L_1^{(\ell)} \sup_{\mathbf{S}_{i,\ell}} \|D_{\mathbf{T}^{(\ell)}}(\mathbf{S}_{i,\ell})\| \right), \quad (23)$$

with the index (ℓ) indicating quantities and constants associated to the layer ℓ .

Proof: See Section J-A2 in the supplementary material. $\square$

This final result highlights that the stability of an AlgNN is inherited from the stability properties associated to the operators in each layer. Each layer of the AlgNN contributes to increase the size of the stability constants, and we can observe that with the appropriate normalization of C_ℓ and B_ℓ , we obtain a stability bound that is in essence the same derived for filters and for the mapping operators in each layer. However, the discriminability power associated to the AlgNN is by far larger than the one related with the operators in the layers and the filters. This is a consequence of the pointwise nonlinearities mapping information between layers. It is important to highlight that the σ_ℓ that map information between layers are identical for commutative and non commutative architectures. However, the way they redistribute spectral information is tied to the nature of the spectral representations. While in commutative architectures such redistribution is always done between spaces of the same dimension – one dimensional representations –, in non

commutative AlgNNs the redistribution of spectral information occurs in general between spaces of different dimension.

VII. NUMERICAL EXPERIMENTS

In order to provide numerical evidence of the stability results derived for AlgNN with non commutative algebras, we consider two architectures: Multigraph Neural Networks (MultiGNN) and Quaternion Convolutional Networks (QCN). For the first, we propose an architecture for learning on multigraphs and show that if the learned filters are penalized to be Integral Lipschitz, the model is much more resilient to perturbations enforced on the graph. This architecture is used for a rating prediction task on the MovieLens-100K dataset [47]. We then apply the quaternion architecture introduced in [14] to an image classification task using the MNIST dataset [48] and observe that model performance quickly diminishes when additive and relative perturbations are applied to the quaternion filters. As we will elaborate in Subsection VII-B, due to the cyclic nature of the generators in the quaternion algebra, no filter in (26) can be integral Lipschitz. As a consequence, no filter can be selected to mitigate the perturbations that severely affect the performance of the QCN.

A. Multigraph Neural Networks

We first consider the application of the Multigraph Neural Network as a movie recommendation system [49]. Using the MovieLens-100K dataset containing 100,000 ratings from 943 users, we model the system as a multigraph. Nodes represent movies which are connected by two classes of edges: one measuring the rating similarity and the other measuring the genre similarity. The signals supported by this multigraph are user ratings, with the task of inferring unseen ratings based on ratings given to other movies. On this task, we carry out stability experiments which measure robustness to estimation error from the training set. We find that Multigraph Neural Networks that are penalized to employ integral Lipschitz filters are most stable to perturbations.

Multigraph Formation: From the 1582 movies available, we use the 200 with the highest number of ratings. The movie *Toy Story* is used as the target movie for predicted ratings. Of users who rated this movie, we define a node signal as the rating 1-5 they gave to each movie or 0 if they did not rate it. These signals are divided with a 90%/10% train/test split. Using the training set, we add rating edge weights between movies using the Pearson correlation coefficient and genre edge weights through the Jaccard similarity of overlapping genres. To appeal to sparsity, the 20 highest weighted edges of both edge classes are kept for each node, and the rest of the edges are removed.

Stability: In contrast with the scenario of the quaternion algebras, multigraph signal models are associated with a very rich algebra: the regular algebra of polynomials with multiple independent variables. This guarantees the existence of subclasses of filters that are Lipschitz and integral Lipschitz. Therefore, it is possible to learn filters – Lipschitz and integral Lipschitz – that can mitigate the effect of deformations on the shift operators. We illustrate this in our experiments analyzing

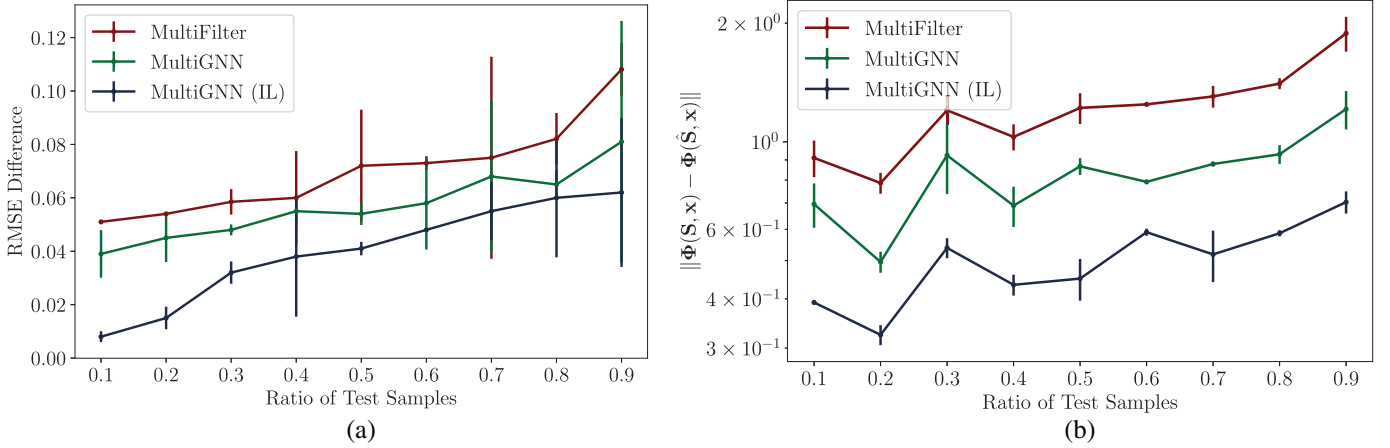


Fig. 6. Stability to perturbations caused by estimation error for the Movie Recommendation problem. We demonstrate in (a) the difference in evaluation measure as the ratio of training samples is increased, where the penalized MultiGNN maintains the smallest difference for all ratios. This trend is also held in (b) where we showcase the difference in the output of the convolutional layer. Then, MultiGNN with IL filters provide more stable and consistent rating predictions with respect to estimation error perturbations.

the magnitude of change in the filters and operators networks when subjected to deformations.

Architectures and Training: In our stability experiments, we consider three architectures. The first, MultiFilter, is a learned linear multigraph filter. We also train a Multigraph Neural Network (MultiGNN) and another which is regularized by an estimate of the filter's integral Lipschitz constant (MultiGNN IL), seeking to train filters that are integral Lipschitz. All architectures employ a convolutional layer with 3 filter taps and 64 output features, a ReLU nonlinearity function, and a local linear readout layer mapping the output rating to a single scalar rating estimate. We minimize the smooth L1 loss using an ADAM optimizer for 40 epochs, and evaluate using the root mean squared error (RMSE). Results are reported as the average and standard deviation of performance across 5 random splits of the dataset.

Estimation Error Experiment: To simulate perturbations as a result of estimation error, we first train each architecture on 90% of the training set. At evaluation, we replace the underlying multigraph support with an estimate generated by a training set ranging from 10% to 90% of the size of the overall dataset. First, we find the difference in RMSE between the trained and evaluated models. The regularized MultiGNN maintains the smallest difference for each training set size as demonstrated in Fig. 6(a). We also compare the norm of the difference in output of the convolutional layer. As can be seen in Fig. 6(b), the MultiGNN with integral Lipschitz filters is approximately an order of magnitude more stable than the unregularized MultiGNN.

B. Quaternion Convolutional Networks

We now consider a non commutative AlgNN proposed in literature: Quaternion Convolutional Networks. Quaternions have shown to be useful tools for modeling spatial transformations with diverse applications to computer graphics, quantum mechanics, and signal processing. Gaudet and Maida [14] propose a deep convolutional architecture for learning through use of quaternion convolutions, quaternion batch-normalization, and

a quaternion weight initialization scheme. We use this architecture as the basis for our second stability experiment.

Architecture: Quaternion convolutions occur through convolving a quaternion filter matrix $\mathbf{W} = \mathbf{A} + i\mathbf{B} + j\mathbf{C} + k\mathbf{D}$ with a quaternion vector $\mathbf{h} = \mathbf{w} + i\mathbf{x} + j\mathbf{y} + k\mathbf{z}$ according to (29) – see Appendix C. In forming a Quaternion Convolutional Network, we compose layers of quaternion convolutions, applying a nonlinear function to the output of each layer. We then append quaternion dense layers, which are traditional fully-connected neural networks applied component-wise. The resultant embedding is shaped into the desired output shape using a readout layer. Our particular architecture is composed of three quaternion convolutional layers (with 32 channels each), two quaternion dense layers (with 82 and 48), and a readout layer, using the ReLU function as the nonlinearity.

Stability: We recall that one of the requirements to guarantee stability to the perturbations in (15) is the existence of subclasses of filters in (26) that are Lipschitz and integral Lipschitz. Although in principle any filter in (26) can be written by considering arbitrary powers on the generators of the algebra, due to the cyclic nature of $\{1, i, j, k\}$, such representations reduce to a linear combination of the generators. Therefore, the subclass of integral Lipschitz filters in (26) is empty. Additionally, while the filters in (26) can still be Lipschitz, the Lipschitz constants will be large. Thus, the performance of the QCN architecture – as defined in [14] – is expected to be severely affected by perturbations, without the potential to design stable filters. We demonstrate this vulnerability in our numerical experiments.

Synthetic Experiment: To corroborate these theoretical results, we first train an unperturbed architecture. Then, during evaluation, we inject noise in the first quaternion convolutional layer by perturbing $\mathbf{A}$, a layer of the quaternion filter matrix $\mathbf{W}_1$. More precisely, we first consider additive perturbations by replacing $\mathbf{A}$ with $\tilde{\mathbf{A}}_1 = \mathbf{A} + \mathbf{T}_1$, where $\mathbf{T}_1$ has its components sampled from the uniform distribution $[-\varepsilon_1, \varepsilon_1]$. We also observe relative perturbations by substituting $\mathbf{A}$ with $\tilde{\mathbf{A}}_2 = \mathbf{A} + \mathbf{T}_2\mathbf{A}$, where $\mathbf{T}_2$ has its values drawn from the uniform distribution $[\varepsilon_2, \varepsilon_2]$.

TABLE I
STABILITY OF QUATERNION CONVOLUTIONAL NETWORKS TO ADDITIVE
AND RELATIVE PERTURBATION

Additive Perturbation				Relative Perturbation			
ε_1	First Con.	Last Con.	Acc.	ε_2	First Con.	Last Con.	Acc.
0.005	27,813	4,415	0.991	0.05	24,986	6,316	0.991
0.01	55,467	9,542	0.991	0.1	49,961	13,100	0.990
0.05	289,610	78,092	0.968	0.5	256,754	79,819	0.956
0.1	604,553	213,046	0.772	1	533,462	158,565	0.825
0.5	3,303,611	234,771	0.222	5	2,975,365	1,361,751	0.321

We apply this architecture to the image classification task using the MNIST dataset [48]. Following [50], we form the quaternion input for each pixel by repeating the grayscale value along each of the three imaginary components and zeroing out the real component. We train the unperturbed architecture for 10 epochs using a batch size of 500. After training, we first evaluate the performance of the unperturbed architecture on the test samples. We then apply additive and multiplicative perturbation to the first quaternion convolutional layer for various values of ε_1 and ε_2 , and observe the new accuracy (Acc.) and norm of the difference between the perturbed and unperturbed architectures of the output from the first quaternion convolutional layer (First Con.) and the last quaternion convolutional layer (Last Con.). Average results across 10 noise samplings are reported in Table 1.

Our results find that Quaternion Convolutional Networks are particularly sensitive to small additive and relative perturbations applied in the first layer, as performance quickly diminishes as ε_1 and ε_2 increases. These results indicate that an architecture trained to strictly minimize loss not result in stable filters.

VIII. DISCUSSION AND CONCLUSIONS

We analyzed non commutative algebraic neural networks (AlgNNs) providing general stability results applicable to convolutional architectures such as multigraph-CNNs, quaternion-CNNs, quiver-CNNs, and Group-NNs (non commutative) among others. We have shown that non commutative convolutional neural networks can be stable. This property has an analogous form to the stability bounds of commutative signal models but with substantial differences. As we showed in previous sections, the restrictions necessary to guarantee stability are defined by algebras of matrices, which highlights the fact that the stability of filters in non commutative signal models is not a trivial extension of the stability results for commutative models. This is a consequence of the fact that spectral representations in non commutative signal models are given by matrix and not scalar functions.

As shown in Theorem 2, the size of the deformation of a filter operator $p(\mathbf{S}_1, \dots, \mathbf{S}_m)$ is bounded by the norm of the Fréchet derivative of p acting on $\mathbf{T}(\mathbf{S}_i)$. This holds for *any arbitrary perturbation* $\mathbf{T}(\mathbf{S}_i)$ indicated in (13) and it highlights the versatility of the deformation model for the representation of a large variety of perturbations.

Notice that non commutative models naturally embed properties of those signal frameworks where there are not shift operator invariances, i.e. $\mathbf{S}_i p(\mathbf{S}_1, \dots, \mathbf{S}_m) \neq p(\mathbf{S}_1, \dots, \mathbf{S}_m) \mathbf{S}_i$. The stability results derived in this paper show that despite the

lack of this invariance, stability is possible and it comes at a price on selectivity and with stronger restrictions on the spectral responses of the filters.

We point out that our results can be extended to more elaborated versions of $\mathbf{T}(\mathbf{S}_i)$. This opens up interesting research applications where $\mathbf{T}(\mathbf{S}_i)$ can be tailored to more specific scenarios where stable filters could be designed for $\mathbf{T}(\mathbf{S}_i) = \sum_r \mathbf{T}_r \mathbf{S}_i^r$, and the restrictions imposed on the algebra will be determined by the properties of the operators $\mathbf{T}_r$.

APPENDIX A BASIC ALGEBRA CONCEPTS

Algebra: We say that the vector space $\mathcal{A}$ is an algebra if $\mathcal{A}$ is equipped with a product. This is, there is an operation of product between two elements of $\mathcal{A}$ that results in another element of $\mathcal{A}$ [35]. Let us denote by ab the product between a and b with $a, b \in \mathcal{A}$. Then, we say that the algebra $\mathcal{A}$ is unital if there exists a multiplicative unit element $\mathbf{1} \in \mathcal{A}$ such that $a\mathbf{1} = \mathbf{1}a = a$ for all $a \in \mathcal{A}$. Typical examples of algebras are:

- $\mathbb{C}$: The set of complex numbers is a vector space over $\mathbb{C}$ itself. Using as algebra product in $\mathbb{C}$ the ordinary product between complex numbers we can see $\mathbb{C}$ as an algebra.
- $\mathbb{C}[t]$: The set of polynomials with coefficients in $\mathbb{C}$ and independent variable t is a vector space over $\mathbb{C}$. Using as algebra product the ordinary product between polynomials we can see $\mathbb{C}[t]$ as an algebra.
- $M_{r \times r}(\mathbb{C})$: The space of matrices of size $r \times r$ with entries in $\mathbb{C}$ is a vector space over $\mathbb{C}$. Using as algebra product the ordinary product between matrices we can see $M_{r \times r}(\mathbb{C})$ as an algebra.

It is important to remark that the formal definition of a *product* is given by a bilinear map. We say that the map $m_{\mathcal{A}} : \mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}$ is bilinear if $m_{\mathcal{A}}$ is linear on each argument. Then, talking about an operation that behaves like a product we talk about a bilinear map that assigns the value of the given product [35].

Endomorphisms: Let $\mathcal{M}$ be a vector space and let $\text{End}(\mathcal{M})$ be the space of linear maps from $\mathcal{M}$ onto itself. If $\dim(\mathcal{M}) = n$, then $\text{End}(\mathcal{M})$ is isomorphic to the space of matrices of size $n \times n$. This is, whenever the dimension of $\mathcal{M}$ is finite we can think about $\text{End}(\mathcal{M})$ as a space of matrices.

APPENDIX B GROUP SIGNAL PROCESSING AND GROUP NEURAL NETWORKS

Classical convolutional information processing on locally compact groups is given by the algebraic model $(\mathcal{A}_G, \mathcal{M}, \rho)$, where $\mathcal{M} = \left\{ \sum_{g \in G} \mathbf{x}(g)g, \quad \mathbf{x}(g) \in \mathbb{C} \right\}$ is the set of complex valued functions defined on the group G . The algebra $\mathcal{A}_G$ is the *group algebra*, which is given by $\mathcal{A}_G = \mathcal{M}$. The homomorphism ρ is given by $\rho(\mathbf{a}) = L_{\mathbf{a}}$ with $L_{\mathbf{a}}\mathbf{x} = \mathbf{a}\mathbf{x}$. Then, the convolution between a group filter and a group signal takes the form

$$\rho \left(\sum_{g \in G} \mathbf{a}(g)g \right) \mathbf{x} = \sum_{g \in G} \mathbf{a}(g)g\mathbf{x} = \sum_{g \in G} \sum_{h \in G} \mathbf{a}(g)\mathbf{x}(h)gh. \quad (24)$$

Notice that the representation of the convolution can also be rewritten making $u = gh$, which leads to [36]

$$\sum_{g,h \in G} \mathbf{a}(g)\mathbf{x}(h)gh = \sum_{u,h \in G} \mathbf{a}(uh^{-1})\mathbf{x}(h)u. \quad (25)$$

We emphasize that the role of this example is to show that classical group convolutions – as considered in [17] – are *one* particular case of algebraic convolutions. What is more, it is one particular case of the many algebraic convolutions that can be associated to the same group algebra. This is, given a non commutative group G and its associated group algebra $\mathcal{A}_G$, it is possible to define different convolutional models $(\mathcal{A}_G, \mathcal{M}, \rho)$ when different choices of $\mathcal{M}$ and ρ are made [15]. This implies that it is possible to leverage the symmetries of the group under a convolutional signal model in different ways and in vector spaces of arbitrary dimension.

Notice that when convolutions are computed as in [17], a *lifting* process is required to map the signals – usually not defined on the group – on the group G . If G is continuous – a Lie group – computing the group convolution requires the approximation of a Haar integral that generalizes (25) to locally compact groups [34]. In [15] the computing of the convolutions – including convolutions with Lie groups – is performed considering an ASM $(\mathcal{A}_G, \mathcal{M}, \rho)$, where $\mathcal{A}_G$ is the group algebra associated to the group G , $\mathcal{M}$ is a vector space of arbitrary dimension and ρ is the homomorphism that will translate approximations of elements in $\mathcal{A}_G$ into concrete operators. In [15] lifting is not necessary since the implementation of the filters in $\mathcal{A}_G$ is carried out by means of an efficient approximation of ρ .

A group convolutional neural network is a stacked layered structure where information in each layer is processed using group convolutions. With convolutions performed as in [15], in each layer of the architecture information is processed according to the triplet $(\mathcal{A}_G, \mathcal{M}_\ell, \rho_\ell)$. The information is mapped between layers by means of $\sigma_\ell = P_\ell \eta_\ell$ where η_ℓ is a point-wise nonlinearity and P_ℓ is a dimensionality reduction operator. When dealing with discrete scenarios we have $\dim(\mathcal{M}) < \infty$ and the action of η_ℓ is expressed in terms of a canonical basis in $\mathbb{R}^{\dim(\mathcal{M})}$.

Notice that the processing of information with [17] can be seen as a particular case of the convolutional architectures proposed in [15] with the advantage that in [15] there is no need for the discretization of the Haar integral since the implementation of the filters relies on building an accurate approximation of ρ_ℓ in $(\mathcal{A}_G, \mathcal{M}_\ell, \rho_\ell)$.

APPENDIX C

QUATERNION SIGNAL PROCESSING AND QUATERNION NEURAL NETWORKS

Let $\mathcal{Q}$ be the algebra on $\mathbb{C}$ generated by the symbols i, j, k and the unit element 1 , where $i^2 = j^2 = k^2 = -1, ij = -ji = k, jk = -kj = i, ki = -ik = j$. $\mathcal{Q}$ is known as the algebra of quaternions [38]. Using $\mathcal{Q}$ is possible to build quaternion-based algebras that can be used to define

convolutional models. In [14] such models are build upon the algebra

$$\mathcal{A} = \{p_1(t)1 + p_2(t)i + p_3(t)j + p_4(t)k \mid 1, i, j, k \in \mathcal{Q}\}, \quad (26)$$

and where $p_i(t) = \sum_{k=0}^K h_{i,k} t^k$ are elements of the algebra of polynomials of a single variable. We can see that $\mathcal{A}$ is a vector space when considering the sum component wise on the quaternion elements, and the multiplication by scalars. The product algebra in $\mathcal{A}$ follows the distribution product of the quaternions. Then, we can define the quaternion convolutional model $(\mathcal{A}, \mathcal{M}, \rho)$, where

$$\mathcal{M} = \{\mathbf{x}_1 1 + \mathbf{x}_2 i + \mathbf{x}_3 j + \mathbf{x}_4 k \mid 1, i, j, k \in \mathcal{Q}; \quad \mathbf{x}_i \in \mathbb{R}^N\}, \quad (27)$$

while the homomorphism ρ is given according to

$$\begin{aligned} \rho(p_1(t)1 + p_2(t)i + p_3(t)j + p_4(t)k) \\ = p_1(\mathbf{C})1 + p_2(\mathbf{C})i + p_3(\mathbf{C})j + p_4(\mathbf{C})k, \end{aligned} \quad (28)$$

where $\rho(t) = \mathbf{C} \in \mathbb{R}^{N \times N}$ is the discrete time delay operator. Then, the convolutional action of a filter $\rho(f) = \mathbf{F}$ on $\mathbf{u} \in \mathcal{M}$ is given by

$$\begin{aligned} \mathbf{F}\mathbf{u} = & (p_1(\mathbf{C})\mathbf{x}_1 - p_2(\mathbf{C})\mathbf{x}_2 - p_3(\mathbf{C})\mathbf{x}_3 - p_4(\mathbf{C})\mathbf{x}_4) 1 \\ & + (p_1(\mathbf{C})\mathbf{x}_2 - p_2(\mathbf{C})\mathbf{x}_1 - p_3(\mathbf{C})\mathbf{x}_4 - p_4(\mathbf{C})\mathbf{x}_3) i \\ & + (p_1(\mathbf{C})\mathbf{x}_3 - p_2(\mathbf{C})\mathbf{x}_4 - p_3(\mathbf{C})\mathbf{x}_1 - p_4(\mathbf{C})\mathbf{x}_2) j \\ & + (p_1(\mathbf{C})\mathbf{x}_4 - p_2(\mathbf{C})\mathbf{x}_3 - p_3(\mathbf{C})\mathbf{x}_2 - p_4(\mathbf{C})\mathbf{x}_1) k. \end{aligned} \quad (29)$$

Notice that $p_i(\mathbf{C})\mathbf{x}_i$ is the ordinary Euclidean convolution in $\mathbb{R}^N$ between a filter and the signal $\mathbf{x}_i$. We emphasize that the particular form of (29) is a consequence of the non commutativity of the filter operators from $\mathcal{A}$ in (26), which is inherited from the quaternion algebra.

A quaternion neural network as introduced in [14] is a stacked layered structure where information is processed in each layer according to (29). The information processed in the layer ℓ has the form $\mathbf{u}_\ell = \mathbf{x}_1 1 + i\mathbf{x}_2 + j\mathbf{x}_3 + k\mathbf{x}_4$, while the convolutional filters are of the form $\mathbf{F} = \mathbf{A}1 + i\mathbf{B} + j\mathbf{C} + k\mathbf{D}$. The action of the filter $\mathbf{F}$ on the signal $\mathbf{u}$ follows the ordinary product between quaternions considering that $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ act as Euclidean convolution operators on $\mathbf{x}_i$ as indicated in (29). Then, given an input $\mathbf{u}_\ell$ to the ℓ -th layer we leverage the symmetries given by the quaternion algebra to obtain $\mathbf{y}_\ell = \mathbf{F}\mathbf{u}_\ell$. After this a point-wise nonlinear operator η_ℓ is applied to obtain $\mathbf{z}_\ell = \eta_\ell(\mathbf{y}_\ell)$ with $\mathbf{z}_\ell(u) = \max\{\mathbf{y}_\ell(u), 0\}$, and where $\mathbf{z}_\ell(u)$ is the u -th component of $\mathbf{z}_\ell$. A conventional pooling operator $P_\ell : \mathbb{R}^N \rightarrow \mathbb{R}^M$ performs traditional sampling on a quaternion signal according to $P_\ell(\mathbf{u}_\ell) = 1P_\ell(\mathbf{x}_1) + iP_\ell(\mathbf{x}_2) + jP_\ell(\mathbf{x}_3) + kP_\ell(\mathbf{x}_4)$.

APPENDIX D

PROOF OF THEOREMS

A. Proof of Theorem 2

Proof: To simplify notation we use $\mathbf{S} = (\mathbf{S}_1, \dots, \mathbf{S}_m)$, and we start taking into account the definition of the Fréchet

derivative

$$p(\mathbf{S} + \boldsymbol{\xi}) = p(\mathbf{S}) + D_p(\mathbf{S}) \{\boldsymbol{\xi}\} + o(\|\boldsymbol{\xi}\|). \quad (30)$$

Taking the norm on both sides of (30) and applying the triangle inequality it follows that

$$\|p(\mathbf{S} + \boldsymbol{\xi}) - p(\mathbf{S})\| \leq \|D_p(\mathbf{S}) \{\boldsymbol{\xi}\}\| + \mathcal{O}(\|\boldsymbol{\xi}\|^2) \quad (31)$$

for all $\boldsymbol{\xi} = (\xi_1, \dots, \xi_m) \in \text{End}(\mathcal{M})^m$. Now, we use the properties of the total derivative (see [51] pages 69-70) to obtain $\|D_p(\mathbf{S}) \{\boldsymbol{\xi}\}\| \leq \sum_{i=1}^m \|D_{p|\mathbf{S}_i}(\mathbf{S}) \{\xi_i\}\|$, which leads to

$$\|p(\mathbf{S} + \boldsymbol{\xi}) - p(\mathbf{S})\| \leq \sum_{i=1}^m \|D_{p|\mathbf{S}_i}(\mathbf{S}) \{\xi_i\}\| + \mathcal{O}(\|\boldsymbol{\xi}\|^2). \quad (32)$$

Then, taking into account that

$$\|p(\mathbf{S} + \boldsymbol{\xi})\mathbf{x} - p(\mathbf{S})\mathbf{x}\| \leq \|\mathbf{x}\| \|p(\mathbf{S} + \boldsymbol{\xi}) - p(\mathbf{S})\| \quad (33)$$

we have

$$\begin{aligned} & \|p(\mathbf{S} + \boldsymbol{\xi})\mathbf{x} - p(\mathbf{S})\mathbf{x}\| \\ & \leq \|\mathbf{x}\| \left(\sum_{i=1}^m \|D_{p|\mathbf{S}_i}(\mathbf{S}) \{\xi_i\}\| + \mathcal{O}(\|\boldsymbol{\xi}\|^2) \right), \end{aligned} \quad (34)$$

and selecting $\xi_i = \mathbf{T}(\mathbf{S}_i)$ we complete the proof. $\square$

B. Proof of Theorem 3

Proof: We start evaluating the expression $\mathbf{D}_{p|\mathbf{S}_i}$ in terms of the specific form of the perturbation to obtain

$$\begin{aligned} \mathbf{D}_{p|\mathbf{S}}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}(\mathbf{S}_i)\} &= \mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{i,0}\} \\ &\quad + \mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{i,1}\mathbf{S}_i\}. \end{aligned} \quad (35)$$

Taking the norm in (35) and using the triangular inequality we have

$$\begin{aligned} & \|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}(\mathbf{S}_i)\}\| \\ & \leq \|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{i,0}\}\| \\ & \quad + \|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{i,1}\mathbf{S}_i\}\|. \end{aligned} \quad (36)$$

Now we analyze each term in (36). First, we start that the Fréchet derivative of $p(\mathbf{S}_1, \dots, \mathbf{S}_m)$ with respect to $\mathbf{S}_i$ acting on $\mathbf{T}_{0,i}$ is a linear operator $\mathbf{D}_{p|\mathbf{S}_i}$ from $\text{End}(\mathcal{M})$ into $\text{End}(\mathcal{M})$ that can be expressed as

$$\mathbf{D}_{p|\mathbf{S}_i} \{\mathbf{T}_{0,i}\} = \sum_{r=1}^{\infty} f_r(\mathbf{S}_1, \dots, \mathbf{S}_m) \mathbf{T}_{0,i} h_r(\mathbf{S}_1, \dots, \mathbf{S}_m), \quad (37)$$

where $f_r(\mathbf{S}_1, \dots, \mathbf{S}_m)$ and $h_r(\mathbf{S}_1, \dots, \mathbf{S}_m)$ are polynomial functions of $\mathbf{S}_1, \dots, \mathbf{S}_m$. As stated in [43] (p. 267), when the operator is assumed to be Hilbert-Schmidt (finite Frobenius norm) there is a unique linear operator $\overline{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\cdot\} \in \text{End}(\mathcal{M})^* \otimes \text{End}(\mathcal{M})$ with $\|\overline{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\cdot\}\| = \|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\cdot\}\|$, acting on a vectorized version of $\mathbf{T}_{0,i}$. In particular we have

$$\overline{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) = \sum_{r=1}^{\infty} h_r^*(\mathbf{S}_1, \dots, \mathbf{S}_m) \otimes f_r(\mathbf{S}_1, \dots, \mathbf{S}_m). \quad (38)$$

Now, we recall that since the representation $(\mathcal{M}, \rho)$ can be expressed as a direct sum of the irreducible subrepresentations. Then, any polynomial from $\mathcal{A}$ mapped into $\text{End}(\mathcal{M})$ by ρ can be expressed in terms of such direct sum by means of the spectral theorem representation (Theorem 1), this is

$$f_r(\mathbf{S}_1, \dots, \mathbf{S}_m) = \sum_{k=1}^q f_r(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k}) \mathcal{P}_k, \quad (39)$$

$$h_r^*(\mathbf{S}_1, \dots, \mathbf{S}_m) = \sum_{k=1}^q h_r^*(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k}) \mathcal{P}_k. \quad (40)$$

Then, substituting (39) and (40) in (38) we have

$$\begin{aligned} \overline{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) &= \sum_{r=1}^{\infty} \left(\sum_{k=1}^q h_r^*(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k}) \mathcal{P}_k \right. \\ &\quad \left. \otimes \sum_{k=1}^q f_r(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k}) \mathcal{P}_k \right). \end{aligned} \quad (41)$$

Taking into account the properties of the tensor product it follows that:

$$\begin{aligned} \overline{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) &= \sum_{k,\ell=1}^q \left(\sum_{r=1}^{\infty} (h_r^*(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k}) \right. \\ &\quad \left. \otimes f_r(\boldsymbol{\Lambda}_{1,\ell}, \dots, \boldsymbol{\Lambda}_{m,\ell})) (\mathcal{P}_k \otimes \mathcal{P}_\ell) \right). \end{aligned} \quad (42)$$

where we have that

$$\begin{aligned} & \sum_{r=1}^{\infty} (h_r^*(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k}) \otimes f_r(\boldsymbol{\Lambda}_{1,\ell}, \dots, \boldsymbol{\Lambda}_{m,\ell})) \\ &= \begin{cases} \overline{\mathbf{D}}_{p|\boldsymbol{\Lambda}_{i,k}}(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k}) & \text{if } k = \ell \\ \mathbf{\Gamma}_{k,\ell} & \text{if } k \neq \ell \end{cases}, \end{aligned} \quad (43)$$

where

$$\begin{aligned} \overline{\mathbf{D}}_{p|\boldsymbol{\Lambda}_{i,k}}(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k}) &= \sum_{r=1}^{\infty} (h_r^*(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k}) \\ &\quad \otimes f_r(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k})), \end{aligned} \quad (44)$$

is uniquely associated to $\mathbf{D}_{p|\boldsymbol{\Lambda}_{i,k}}(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k})$, which is the Fréchet derivative of $p(\boldsymbol{\Lambda}_{1,k}, \dots, \boldsymbol{\Lambda}_{m,k})$ with respect to $\boldsymbol{\Lambda}_{i,k}$. Additionally, we point that the terms in the sum of (42)

are unique, and consequently can be expressed as a direct sum. This is,

$$\begin{aligned} \overline{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) &= \bigoplus_{k,\ell=1}^q \sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \\ &\quad \otimes f_r(\Lambda_{1,\ell}, \dots, \Lambda_{m,\ell})) (\mathcal{P}_k \otimes \mathcal{P}_\ell). \end{aligned} \quad (45)$$

Then, if we calculate the norm of $\overline{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m)$ and we take into account the *maximum property* (Definition 12), it follows that

$$\|\overline{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m)\| = \max_{(k,\ell)} \{\|\overline{\mathbf{D}}_{p|\Lambda_{i,k}}\|, \|\Gamma_{k,\ell}\|\}. \quad (46)$$

First, we recall that $\|\overline{\mathbf{D}}_{p|\Lambda_{i,k}}\| = \|\mathbf{D}_{p|\Lambda_{i,k}}\|$ and since p is L_0 -Lipschitz by Theorem 8 we have that $\|\overline{\mathbf{D}}_{p|\Lambda_{i,k}}\| \leq L_0$. We focus now on estimating the norm of $\Gamma_{k,\ell}$. First, we recall that

$$\Gamma_{k,\ell} = \sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \otimes f_r(\Lambda_{1,\ell}, \dots, \Lambda_{m,\ell})), \quad (47)$$

with $k \neq \ell$. Now, we notice that it is possible to define a homomorphism θ between the algebras of operators $\text{End}(\mathcal{U}_k)^* \otimes \text{End}(\mathcal{U}_\ell)$ and $\text{End}(\mathcal{U}_k)^* \otimes \text{End}(\mathcal{U}_\ell)$ with $\ell < k$, which is given by

$$\begin{aligned} \theta \left(\sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \otimes f_r(\Lambda_{1,k}, \dots, \Lambda_{m,k})) \right) \\ = \sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \otimes f_r(\Lambda_{1,\ell}, \dots, \Lambda_{m,\ell})). \end{aligned} \quad (48)$$

By means of Theorem 7 it follows that $\|\Gamma_{k,\ell}\| \leq \|\overline{\mathbf{D}}_{p|\Lambda_{i,k}}\|$, and therefore leads to $\|\Gamma_{k,\ell}\| \leq L_0$. Then, with the calculations above we have $\|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m)\| \leq L_0$. We recall that for bounded operators we have

$$\begin{aligned} \|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{0,i}\}\| \\ \leq \|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{0,i}\}\|_F, \end{aligned}$$

and as stated in [43] (p. 267) we also have that

$$\begin{aligned} \|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{0,i}\}\|_F \\ \leq \|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m)\| \|\mathbf{T}_{0,i}\|_F. \end{aligned}$$

Then, taking into account that $\|\mathbf{T}_{0,i}\|_F \leq \delta \|\mathbf{T}_{0,i}\|$, we reach

$$\|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{0,i}\}\| \leq L_0 \delta \sup_{\mathbf{S}_i} \|\mathbf{T}(\mathbf{S}_i)\|. \quad (49)$$

Now, we turn our attention to the term $\|\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{1,i}\}\|$ in (36). We start using the notation

$$\mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{1,i}\} = \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\mathbf{T}_{1,i}\}. \quad (50)$$

Now, we point out that

$$\begin{aligned} \widetilde{\mathbf{D}}_{p|\mathbf{S}_i} \{\mathbf{T}_{1,i}\} &= \sum_{r=1}^{\infty} f_r(\mathbf{S}_1, \dots, \mathbf{S}_m) \mathbf{T}_{1,i} h_r(\mathbf{S}_1, \dots, \mathbf{S}_m), \\ &\quad (51) \end{aligned}$$

where $f_r(\mathbf{S}_1, \dots, \mathbf{S}_m)$ and $h_r(\mathbf{S}_1, \dots, \mathbf{S}_m)$ are polynomial functions of $\mathbf{S}_1, \dots, \mathbf{S}_m$. When the operator is assumed to be Hilbert-Schmidt there is a unique linear operator given by $\widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\cdot\} \in \text{End}(\mathcal{M})^* \otimes \text{End}(\mathcal{M})$ with $\|\widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\cdot\}\| = \|\widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{\cdot\}\|$, acting on a vectorized version of $\mathbf{T}_{1,i}$ [43] (p. 267). In particular we have

$$\widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) = \sum_{r=1}^{\infty} h_r^*(\mathbf{S}_1, \dots, \mathbf{S}_m) \otimes f_r(\mathbf{S}_1, \dots, \mathbf{S}_m). \quad (52)$$

Since $(\mathcal{M}, \rho)$ can be expressed as a direct sum of the irreducible subrepresentations. Then, any polynomial from $\mathcal{A}$ mapped into $\text{End}(\mathcal{M})$ by ρ can be expressed in terms of such direct sum by means of the spectral theorem representation (Theorem 1), this is

$$f_r(\mathbf{S}_1, \dots, \mathbf{S}_m) = \sum_{k=1}^q f_r(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \mathcal{P}_k, \quad (53)$$

$$h_r^*(\mathbf{S}_1, \dots, \mathbf{S}_m) = \sum_{k=1}^q h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \mathcal{P}_k. \quad (54)$$

Then, substituting (53) and (54) in (52) we have

$$\begin{aligned} \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) &= \sum_{r=1}^{\infty} \left(\sum_{k=1}^q h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \mathcal{P}_k \right. \\ &\quad \left. \otimes \sum_{k=1}^q f_r(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \mathcal{P}_k \right). \end{aligned}$$

Taking into account the properties of the tensor product it follows that:

$$\begin{aligned} \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) &= \sum_{k,\ell=1}^q \left(\sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \right. \\ &\quad \left. \otimes f_r(\Lambda_{1,\ell}, \dots, \Lambda_{m,\ell})) (\mathcal{P}_k \otimes \mathcal{P}_\ell) \right), \end{aligned} \quad (55)$$

where we have that

$$\begin{aligned} \sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \otimes f_r(\Lambda_{1,\ell}, \dots, \Lambda_{m,\ell})) \\ = \begin{cases} \widetilde{\mathbf{D}}_{p|\Lambda_{i,k}}(\Lambda_{1,k}, \dots, \Lambda_{m,k}) & \text{if } k = \ell \\ \Gamma_{k,\ell} & \text{if } k \neq \ell \end{cases}, \end{aligned} \quad (56)$$

with

$$\begin{aligned} \widetilde{\mathbf{D}}_{p|\Lambda_{i,k}}(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \\ = \sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \otimes f_r(\Lambda_{1,k}, \dots, \Lambda_{m,k})), \end{aligned} \quad (57)$$

is uniquely associated to $\mathbf{D}_{p|\Lambda_{i,k}}(\Lambda_{1,k}, \dots, \Lambda_{m,k})\{(\cdot)_{\Lambda_{i,k}}\}$. Like in the scenarios discussed above, we point that the

terms in the sum of (55) are unique, and consequently can be expressed as a direct sum as follows

$$\begin{aligned} \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) &= \bigoplus_{k,\ell=1}^q \sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \\ &\quad \otimes f_r(\Lambda_{1,\ell}, \dots, \Lambda_{m,\ell})) (\mathcal{P}_k \otimes \mathcal{P}_\ell). \end{aligned} \quad (58)$$

If we calculate the norm of $\widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m)$ and we use the *maximum property* (Definition 12), it follows that

$$\left\| \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \right\| = \max_{(k,\ell)} \left\{ \left\| \widetilde{\mathbf{D}}_{p|\Lambda_{i,k}} \right\|, \left\| \Gamma_{k,\ell} \right\| \right\}. \quad (59)$$

Since $\left\| \widetilde{\mathbf{D}}_{p|\Lambda_{i,k}} \right\| = \left\| \widetilde{\mathbf{D}}_{p|\Lambda_{i,k}} \right\|$, and p is L_1 -integral Lipschitz, we have that $\left\| \widetilde{\mathbf{D}}_{p|\Lambda_{i,k}} \right\| \leq L_1$. Now we analyze the norm of $\Gamma_{k,\ell}$. First, we recall that

$$\Gamma_{k,\ell} = \sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \otimes f_r(\Lambda_{1,\ell}, \dots, \Lambda_{m,\ell})), \quad (60)$$

with $k \neq \ell$. We point out that it is possible to define a homomorphism θ between the algebras of operators $\text{End}(\mathcal{U}_k)^* \otimes \text{End}(\mathcal{U}_k)$ and $\text{End}(\mathcal{U}_k)^* \otimes \text{End}(\mathcal{U}_\ell)$ with $k < \ell$, which is given by

$$\begin{aligned} \theta \left(\sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \otimes f_r(\Lambda_{1,k}, \dots, \Lambda_{m,k})) \right) \\ = \sum_{r=1}^{\infty} (h_r^*(\Lambda_{1,k}, \dots, \Lambda_{m,k}) \otimes f_r(\Lambda_{1,\ell}, \dots, \Lambda_{m,\ell})). \end{aligned} \quad (61)$$

By Theorem 7, it follows that $\left\| \Gamma_{k,\ell} \right\| \leq \left\| \widetilde{\mathbf{D}}_{p|\Lambda_{i,k}} \right\|$, which leads to $\left\| \Gamma_{k,\ell} \right\| \leq L_1$, and therefore $\left\| \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \right\| \leq L_1$. If $\widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m)$ is a bounded operator, we have

$$\begin{aligned} \left\| \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{ \mathbf{T}_{1,i} \} \right\| \\ \leq \left\| \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{ \mathbf{T}_{1,i} \} \right\|_F, \end{aligned}$$

and as stated in [43] (p. 267) we also have that

$$\begin{aligned} \left\| \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{ \mathbf{T}_{1,i} \} \right\|_F \\ \leq \left\| \widetilde{\mathbf{D}}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \right\| \left\| \mathbf{T}_{1,i} \right\|_F. \end{aligned}$$

Taking into account that $\left\| \mathbf{T}_{1,i} \right\|_F \leq \delta \left\| \mathbf{T}_{1,i} \right\|$, we reach

$$\left\| \mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{ \mathbf{T}_{1,i} \mathbf{S}_i \} \right\| \leq L_1 \delta \left\| \mathbf{T}_{1,i} \right\|, \quad (62)$$

and since $\mathbf{D}_{\mathbf{T}}(\mathbf{S}_i) = \mathbf{T}_{1,i}$, this leads to

$$\left\| \mathbf{D}_{p|\mathbf{S}_i}(\mathbf{S}_1, \dots, \mathbf{S}_m) \{ \mathbf{T}_{1,i} \mathbf{S}_i \} \right\| \leq L_1 \delta \sup_{\mathbf{S}_i} \left\| \mathbf{D}_{\mathbf{T}}(\mathbf{S}_i) \right\|. \quad (63)$$

□

REFERENCES

- [1] A. Parada-Mayorga and A. Ribeiro, "Algebraic neural networks: Stability to deformations," *IEEE Trans. Signal Process.*, vol. 69, pp. 3351–3366, 2021.
- [2] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 5680–5695, 2020.
- [3] F. Gama, J. B. Estrach, and A. Ribeiro, "Diffusion scattering transforms on graphs," in *Proc. 7th Int. Conf. Learn. Representations (ICLR)*, May 6–9, 2019, pp. 1–12.
- [4] S. Mallat, "Group invariant scattering," *Commun. Pure Appl. Math.*, vol. 65, no. 10, pp. 1331–1398, 2012.
- [5] M. Püschel and J. M. F. Moura, "Algebraic signal processing theory," 2006, *arXiv:cs/0612077*.
- [6] A. Parada-Mayorga and A. Ribeiro, "Stability of algebraic neural networks to small perturbations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 5205–5209.
- [7] B. Knyazev, X. Lin, M. R. Amer, and G. W. Taylor, "Spectral multigraph networks for discovering and fusing relationships in molecules," 2018, *arXiv:1811.09595*.
- [8] M. Weiler and G. Cesa, "General E(2)-equivariant steerable CNNs," 2019, *arXiv:1911.08251*.
- [9] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. Cohen, "3D steerable CNNs: Learning rotationally equivariant features in volumetric data," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–12.
- [10] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 7168–7177.
- [11] M. Finzi, S. Stanton, P. Izmailov, and A. G. Wilson, "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 3165–3176.
- [12] M. Finzi, M. Welling, and A. G. Wilson, "A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups," 2021, *arXiv:2104.09459*.
- [13] J. Hoffmann, S. Schmitt, S. Osindero, K. Simonyan, and E. Elsen, "Algebranets," 2020, *arXiv:2006.07360*.
- [14] C. J. Gaudet and A. S. Maida, "Deep quaternion networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*. Piscataway, NJ, USA: IEEE, 2018, pp. 1–8.
- [15] H. Kumar, A. Parada-Mayorga, and A. Ribeiro, "Lie group algebra convolutional filters," 2023, *arXiv:2305.04431*.
- [16] L. Butler, A. Parada-Mayorga, and A. Ribeiro, "Convolutional learning on multigraphs," *IEEE Trans. Signal Process.*, vol. 71, pp. 933–946, 2023.
- [17] M. Michael, J. Bronstein, T. Bruna, P. Cohen, and Velivckovi, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," 2021, *arXiv:2104.13478*.
- [18] M. Weiler, P. Forré, E. Verlinde, and M. Welling, "Coordinate independent convolutional networks-isometry and gauge equivariant convolutions on Riemannian manifolds," 2021, *arXiv:2106.06020*.
- [19] L. Butler, A. Parada-Mayorga, and A. Ribeiro, "Learning with multigraph convolutional filters," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.
- [20] T. D. Dai Quoc Nguyen, D. Q. Nguyen, and Phung, "Quaternion graph neural networks," in *Proc. 13th Asian Conf. Mach. Learn. (ACML)*, 2020, pp. 236–251.
- [21] T. Parcollet et al., "Quaternion recurrent neural networks," 2018, *arXiv:1806.04418*.
- [22] X. Qiu, T. Parcollet, M. Ravanelli, N. D. Lane, and M. Morchid, "Quaternion neural networks for multi-channel distant speech recognition," in *Proc. INTERSPEECH*, 2020, pp. 329–333.
- [23] Y. Xia, M. Xiang, Z. Li, and D. P. Mandic, "Echo state networks for multidimensional data: Exploiting noncircularity and widely linear models," in *Adaptive Learning Methods for Nonlinear System Modeling*. Amsterdam, The Netherlands: Elsevier, 2018, pp. 267–288.
- [24] T. Parcollet, M. Morchid, and G. Linares, "A survey of quaternion neural networks," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2957–2982, 2020.
- [25] E. Grassucci, E. Cicero, and D. Comminiello, "Quaternion generative adversarial networks," in *Generative Adversarial Learning: Architectures and Applications*. Cham: Springer, 2022, pp. 57–86.
- [26] E. Grassucci, A. Zhang, and D. Comminiello, "PHNNs: Lightweight neural networks via parameterized hypercomplex convolutions," *IEEE*

- Trans. Neural Netw. Learn. Syst.*, early access, Dec. 13, 2022, doi: [10.1109/TNNLS.2022.3226772](https://doi.org/10.1109/TNNLS.2022.3226772).
- [27] S. Buchholz and G. Sommer, "A hyperbolic multilayer perceptron," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. (IJCNN)*. *Neural Comput.: New Challenges Perspect. New Millennium*, vol. 2, 2000, pp. 129–133.
 - [28] T. Nitta and S. Buchholz, "On the decision boundaries of hyperbolic neurons," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*. Piscataway, NJ, USA: IEEE, 2008, pp. 2974–2980.
 - [29] C.-A. Popa, "Octonion-valued neural networks," in *Proc. Int. Conf. Artif. Neural Netw.* Cham: Springer, 2016, pp. 435–443.
 - [30] E. Bayro-Corrochano, R. Vallejo, and N. Arana-Daniel, "Geometric preprocessing, geometric feedforward neural networks and Clifford support vector machines for visual learning," *Neurocomputing*, vol. 67, pp. 54–105, 2005.
 - [31] S. Buchholz and G. Sommer, "On Clifford neurons and Clifford multi-layer perceptrons," *Neural Netw.*, vol. 21, no. 7, pp. 925–935, 2008.
 - [32] S. Buchholz and G. Sommer, "Clifford algebra multilayer perceptrons," in *Geometric Computing with Clifford Algebras*. Berlin, Germany: Springer, 2001, pp. 315–334.
 - [33] A. Parada-Mayorga, H. Riess, A. Ribeiro, and R. Ghrist, "Quiver signal processing (QSP)," 2010, *arXiv:2010.11525*.
 - [34] G. B. Folland, *A Course in Abstract Harmonic Analysis. Textbooks in Mathematics*. Boca Raton, FL, USA: CRC Press, 2016.
 - [35] M. Lorenz, *A Tour of Representation Theory. Graduate Studies in Mathematics*. Providence, RI, USA: AMS, 2018.
 - [36] A. Terras, *Fourier Analysis on Finite Groups and Applications*. New York, NY, USA: Cambridge Univ. Press, 1999.
 - [37] P. Diaconis and P. W. Diaconis, *Group Representations in Probability and Statistics* (IMS Lecture Notes). Hayward, CA, USA: Institute of Mathematical Statistics, 1988.
 - [38] P. I. Etingof et al., *Introduction to Representation Theory* (Student Mathematical Library). Providence, RI, USA: AMS, 2011.
 - [39] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 4, pp. 1034–1049, Feb. 2019.
 - [40] Y.-L. Boureau, J. Ponce, and Y. Lecun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*. Madison, WI, USA: Omnipress, 2010, pp. 111–118.
 - [41] Y.-L. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. Lecun, "Ask the locals: Multi-way local pooling for image recognition," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2651–2658.
 - [42] A. Parada-Mayorga, Z. Wang, and A. Ribeiro, "Graphon pooling for reducing dimensionality of signals and convolutional operators on graphs," 2022, *arXiv:2212.08171*.
 - [43] J. B. Conway, *A Course in Functional Analysis* (Graduate Texts in Mathematics). New York, NY, USA: Springer, 1994.
 - [44] M. W. Hirsch, *Differential Topology* (Graduate Texts in Mathematics). New York, NY, USA: Springer, 1997.
 - [45] J. A. Leslie, "On a differential structure for the group of diffeomorphisms," *Topology*, vol. 6, no. 2, pp. 263–271, 1967.
 - [46] A. Banyaga, *The Structure of Classical Diffeomorphism Groups. Mathematics and Its Applications*. New York, NY, USA: Springer, 2013.
 - [47] F. M. Harper and J. A. Konstan, "The MovieLens datasets," *ACM Trans. Interactive Intell. Syst.*, vol. 5, no. 4, pp. 1–19, 2016.
 - [48] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
 - [49] W. Huang, A. G. Marques, and A. R. Ribeiro, "Rating prediction via graph signal processing," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5066–5081, Oct. 2018.
 - [50] T. Parcollet, M. Morchid, and G. Linares, "Quaternion convolutional neural networks for heterogeneous image processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. Piscataway, NJ, USA: IEEE, 2019, pp. 8514–8518.
 - [51] M. S. Berger, *Nonlinearity and Functional Analysis: Lectures on Nonlinear Problems in Mathematical Analysis* (Pure and Applied Mathematics). Amsterdam, The Netherlands: Elsevier, 1977.
 - [52] P. Lancaster and H. K. Farahat, "Norms on direct sums and tensor products," *Math. Comput.*, vol. 26, no. 118, pp. 401–414, 1972.
 - [53] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge Univ. Press, 2012.
 - [54] K. Lux and H. Pahlings, *Representations of Groups: A Computational Approach* (Cambridge Studies in Advanced Mathematics). New York, NY, USA: Cambridge Univ. Press.
 - [55] P. Dräxler, G. Michler, and C. M. Ringel, *Computational Methods for Representations of Groups and Algebras: Euroconference in Essen (Germany), April 1–5, 1977* (Progress in Mathematics). Basel: Birkhauser, 2012.
 - [56] M. Barot, *Introduction to the Representation Theory of Algebras*. Cham: Springer, 2014.