Transferability of coVariance Neural Networks

Saurabh Sihag, Gonzalo Mateos, *Senior Member, IEEE*, Corey McMillan, and Alejandro Ribeiro, *Senior Member, IEEE*

***Abstract*—Graph convolutional networks (GCN) leverage topology-driven graph convolutional operations to combine information across the graph for inference tasks. In our recent work, we have studied GCNs with covariance matrices as graphs in the form of coVariance neural networks (VNNs) and shown that VNNs draw similarities with traditional principal component analysis (PCA) while overcoming its limitations regarding instability. In this paper, we focus on characterizing the transferability of VNNs. The notion of transferability is motivated from the intuitive expectation that learning models could generalize to "compatible" datasets (i.e., datasets of different dimensionalities describing the same domain) with minimal effort. VNNs inherit the scale-free data processing architecture from GCNs and here, we show that VNNs exhibit transferability of performance (without re-training) over datasets whose covariance matrices converge to a limit object. Multi-scale neuroimaging datasets enable the study of the brain at multiple scales and hence, provide an ideal scenario to validate the transferability of VNNs. We first demonstrate the quantitative transferability of VNNs over a regression task of predicting chronological age from a multi-scale dataset of cortical thickness features. Further, to elucidate the advantages offered by VNNs in neuroimaging data analysis, we also deploy VNNs as regression models in a pipeline for "brain age" prediction from cortical thickness features. The discordance between brain age and chronological age ("brain age gap") can reflect increased vulnerability or resilience toward neurological disease or cognitive impairments. The architecture of VNNs allows us to extend beyond the coarse metric of brain age gap and associate anatomical interpretability to elevated brain age gap in Alzheimer's disease (AD). We leverage the transferability of VNNs to cross validate the anatomical interpretability offered by VNNs to brain age gap across datasets of different dimensionalities.**

***Index Terms*—graph convolutional network, principal component analysis, transferability, graphon, brain age, interpretability.**

I. Introduction

In various modern applications, the number of features (denoted by m) in a dataset is a fundamental component of data acquisition that is typically a characteristic of the desired application and logistics involved [1], [2]. Most machine learning algorithms and statistical inference approaches are designed over a pre-defined feature set and hence, their computational and sample complexities inherently depend on the dimensionality m [3], [4]. In this paper, we study a deep learning framework called coVariance neural networks (VNN) [5] that is based on graph neural networks (GNNs) operating on the sample covariance matrix from a given dataset and is scale-free, i.e., the number of learnable parameters in VNN is independent of the dimensionality m of the dataset. The scale-free aspect of VNNs makes it feasible for them to be *transferable* between datasets of different dimensionalities without any changes to their architecture, i.e., VNNs trained on a dataset with dimensionality $m = m_1$ can process another dataset with dimensionality $m = m_2$ with the same set of learned parameters. Thus, the notion of transferability of VNNs in this paper is primarily focused on transferability across datasets of different dimensionalities, under the same domain. Other notions of transferability in machine learning, such as domain transferability or transference between datasets of different distributions, are not considered in this paper.

The convolution operation in VNNs is modeled by a polynomial coVariance filter over a sample covariance matrix [5]. Covariance matrices and principal component analysis (PCA) form the foundations of non-parametric analyses in various practical applications that are characterized by spatially distributed, multi-variate data acquisition protocols. Some examples of such applications include neuroimaging [6], computer vision [7], weather modeling [8], traffic flow analysis [9], and cloud computing [10]. Moreover, GCNs admit the properties of stability to topological perturbations and transferability across graphs of different sizes in various settings [11], [12], [13], [14], which makes them a well-motivated data analysis tool for graph-structured data. Our results in [5] formalized the following significant observations: i) there exist similarities between the spectral analysis of graph convolution on a covariance matrix and the standard PCA transformation; and ii) VNNs are robust to the number of samples used to estimate the sample covariance matrix, thus, overcoming a potential source of instability and irreproducibility of PCA based statistical inference [15], [16].

The transferability of GNNs from training graphs to some compatible family of test graphs has been previously studied from different theoretical perspectives [14], [17], [18], [19]. The notion of transferability of GNNs broadly encapsulates the intuition that GNNs may be able to retain their performance for some inference task when applied over test graphs (irrespective

Manuscript received 10 July 2023; revised 4 March 2024; accepted 4 March 2024. Date of publication 28 March 2024; date of current version 3 July 2024. This work was supported by NSF CCF under Grant 1934960. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Aylin Yener. *(Corresponding author: Saurabh Sihag.)*

Saurabh Sihag and Alejandro Ribeiro are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: sihag.saurabh@gmail.com).

Gonzalo Mateos is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA.

Corey McMillan is with the Department of Neurology, University of Pennsylvania, Philadelphia, PA 19104 USA.

This article has supplementary downloadable material available at https://doi.org/10.1109/JSTSP.2024.3378887, provided by the authors.

Digital Object Identifier 10.1109/JSTSP.2024.3378887

of the size) that describe the same phenomenon as the training graphs. In this context, the study in [17] considers transferability of GNNs over graphs that represent the discretization of underlying topological spaces. Several studies also consider GNN transferability over graphs that belong to a converging sequence that approaches a limiting object in the asymptote of a large number of nodes [14], [19]. In [18], the similarity between the ego graph distributions (derived from graph topology) formed the workhorse for assessing transferability of GNNs. Transferability of GNNs also provides advantages in terms of computational complexity, which for GNNs scales as $\mathcal{O}(m^2)$ for dense graphs with m nodes. In this paper, we broaden the notion of transferability to VNNs and establish the transference over covariance matrices of different sizes that converge in some sense. In this context, transferability is not feasible for traditional statistical models as they are restricted within the feature space of the original dataset and need to be re-evaluated if the number of features change. Thus, transferability of VNNs is broadly relevant to the domain of multivariate statistics.

Neuroimaging is an example of a timely application in which the number of features can vary across datasets, yet different datasets contain similar information [20], [21]. Specifically, MRI data can be represented in many scales ranging from single voxels (typically ~ 1 mm^3) to regions-of-interest (ROIs) derived from multi-scale brain atlases that range from dozens to thousands of parcellations (e.g., from 100 to 1000 number of parcellations in a multi-scale brain atlas [22], [23]). There has been a growing interest in multi-scale datasets in neuroscience [21], [24], [25], [26], [27]. These datasets rely on brain atlases or templates that allow a multi-scale parcellation of the brain surface (for instance, Schaefer's atlas [22] and Lausanne atlas [23]). A multi-scale brain atlas partitions the brain cortex into a variable number of regions at different scales. However, statistically sound approaches that can leverage or cross-validate the redundancy of information in datasets at multiple scales are currently lacking.

Our recent work has demonstrated that VNNs can provide an anatomically interpretable perspective to the task of "brain age" prediction from cortical thickness features in AD [28]. Accelerated aging (i.e., when brain age estimated from neuroimaging is elevated as compared to chronological age) can be a predictor of cognitive decline or neurological conditions like Alzheimer's disease and related dementias (ADRD) [29], [30]. Hence, in this application, the difference between the brain age estimated from neuroimaging data and the chronological age, i.e., the *brain age gap* (Δ-Age) is the metric of interest. Here, we leverage the transferability of VNNs to cross-validate Δ-Age predictions and their associated anatomical interpretability across multi-scale datasets.

Several machine learning and statistical approaches have been adopted to estimate brain age from neuroimaging data [29], [30], [31], [32], [33], [34], [35], [36]. Commonly, these methods rely on models trained to predict chronological age of healthy population. Deep learning approaches are often adopted due to their ability to provide highly accurate estimates of chronological age in healthy population. However, the accuracy of chronological age prediction in healthy population may not be correlated with the clinical utility of brain age estimates in adverse health conditions [37]. Moreover, due to lack of transparency in such models, it is not guaranteed that relevant disease-driven effects in the neuroimaging data were indeed leveraged in the estimates of brain age. To address this, limited studies have utilized state-of-the-art post-hoc, model-agnostic methods such as SHAP or LIME [36] and saliency maps [35] to add explainability to their brain age estimation approaches, identifying the input features most relevant to the inference outcome. However, explainability offered by such approaches may be unstable to small perturbations to the input [38], [39], inconsistent to variations in training algorithms [40] and model multiplicity (i.e., when multiple models with similar performance may exist) [41], and computationally expensive [42]. In this context, VNNs provide a *transparent* learning model that is inherently interpretable and can associate elevated Δ-Age with brain regions characteristic of a disease or health condition as well as the principal components of the covariance matrix, with no significant added computational cost. Importantly, the implication of the transferability of VNNs is that the anatomical interpretability offered by VNNs can also be guaranteed on datasets of different dimensionalities; a feature which is infeasible for state-of-the-art explainability methods.

Contributions: Our contributions can be summarized as follows:

- *Transferability of VNNs:* We theoretically characterize the transferability of VNNs between datasets of different dimensionalities. For a dataset with m_1 features and covariance matrix $\mathbf{C}_{m_1}$ and another dataset with m_2 features and covariance matrix $\mathbf{C}_{m_2}$, we establish that the outputs of a VNN when instantiated on $\mathbf{C}_{m_1}$ and $\mathbf{C}_{m_2}$ are close in some sense under appropriate conditions on the covariance matrices $\mathbf{C}_{m_1}$ and $\mathbf{C}_{m_2}$ (see Theorem 2).
- *Evaluation on multi-scale neuroimaging datasets:* We train VNNs for the regression task of predicting chronological age from cortical thickness features. The transferability of VNNs is validated by the transference of regression performance across multi-scale cortical thickness datasets curated according to different scales of a commonly used brain atlas (Section IV-B). Further, in Section IV-C we deploy VNNs in the pipeline for anatomically interpretable brain age prediction from [28] and compare the Δ-Age between healthy controls and individuals with AD diagnosis. We leverage the transferability of VNNs to cross-validate the distributions of Δ-Age and the accompanying anatomical interpretability across cortical thickness datasets available at different scales of a multi-resolution brain atlas. In the interest of reproducibility, the code developed is publicly available at https://github.com/sihags/VNN_Brain_Age.

Previously, we have empirically demonstrated transferability of VNNs on a regression task in [5] and a brain age prediction task in [43]. However, no theoretical results regarding transferability were provided in these preliminary studies. Moreover, our prior work in [43] did not associate brain age with anatomical interpretability. This paper extends the contributions in [5] and [43] in various significant ways. First, we develop a theoretical framework to establish transferability of VNNs.

Also, a more comprehensive empirical evaluation in the context of brain age is provided, where we leverage the transferability of VNNs to cross-validate anatomical interpretability across datasets of different dimensionalities.

II. Covariance Neural Networks

VNNs operate on covariance matrices and have similar architecture as GNNs.[1] We start by providing preliminary definitions pertaining to the architecture and discuss the theoretical properties associated with VNNs later.

Consider an $m-$dimensional random vector $\mathbf{x} \in \mathbb{R}^{m\times 1}$ whose ensemble covariance matrix is defined as

$$\mathbf{C} \triangleq \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^{\mathsf{T}}], \tag{1}$$

where $\cdot^{\mathsf{T}}$ is the transpose operator and $\mathbb{E}[\cdot]$ is the expectation with respect to the probability distribution of $\mathbf{x}$. In practice, we usually have access to a dataset that provides us with the statistical information about $\mathbf{x}$. Therefore, we also consider a dataset consisting of n random, independent and identically distributed (i.i.d) samples of $\mathbf{x}$, given by $\mathbf{x}_i \in \mathbb{R}^{m\times 1}, \forall i \in \{1,\ldots,n\}$, where the dataset can be represented in matrix form as $\mathbf{X}_n = [\mathbf{x}_1,\ldots,\mathbf{x}_n]$. Using $\mathbf{X}_n$, the sample covariance matrix estimator is given by

$$\hat{\mathbf{C}} \triangleq \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^{\mathsf{T}}, \tag{2}$$

where $\bar{\mathbf{x}}$ is the sample mean. Next, we discuss the motivation behind studying VNNs separately from GNNs.

A. Motivation

Covariance matrices are ubiquitous in real world applications that have spatially distributed, multi-variate data acquisition protocols [6], [8], [9], [10]. The eigenvectors of covariance matrices are termed as principal components of the dataset and constitute the PCA transformation [44]. The covariance matrix $\mathbf{C}$ can be viewed as the adjacency matrix of a graph representing the stochastic structure of the vector $\mathbf{x}$, where the m dimensions of $\mathbf{x}$ can be thought of as the nodes of an m-node, undirected graph and its edges represent the pairwise covariance between elements in $\mathbf{x}$. Furthermore, the eigenvalues of $\mathbf{C}$ encode the variability of the dataset along different directions in an orthogonal space determined by the associated eigenvectors or principal components.

In graph signal processing, a vector defined on the nodes of the graph is viewed as the graph signal and the projection of a graph signal on the eigenbasis of the graph yields the graph Fourier transform [45]. The graph Fourier transform provides a systematic mathematical tool to analyze convolutional filters over graphs [46], [47]. Interestingly, the classical Fourier transform and graph Fourier transform converge over a discrete, periodic time series represented on a directed, cyclic graph [48]. Similarly to the graph Fourier transform, we can define the coVariance Fourier transform as the projection of a random instance $\mathbf{x}$[2] on the eigenvectors of the covariance matrix $\mathbf{C}$ [5, Definition 1]. The definition of coVariance Fourier transform from [5] is stated next. For this purpose, consider the eigendecomposition of $\mathbf{C}$ given by

$$\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathsf{T}}, \tag{3}$$

where $\mathbf{V} = [\mathbf{v}_1,\ldots,\mathbf{v}_m]$ is a matrix of size $m \times m$ with its columns as the eigenvectors and $\mathbf{\Lambda} = \mathsf{diag}(\lambda_1,\ldots,\lambda_m)$ is a diagonal matrix with its diagonal elements representing the eigenvalues of $\mathbf{C}$.

Definition 1 (coVariance Fourier Transform): The coVariance Fourier transform (VFT) of a random sample $\mathbf{x}$ is defined as its projection on the eigenspace of $\mathbf{C}$ and is given by

$$\tilde{\mathbf{x}} \triangleq \mathbf{V}^{\mathsf{T}}\mathbf{x}. \tag{4}$$

The i-th entry of $\tilde{\mathbf{x}}$, i.e., $[\tilde{\mathbf{x}}]_i$ represents the projection of $\mathbf{x}$ on eigenvector $\mathbf{v}_i$ and hence, it is associated with the eigenvalue λ_i. Thus, the similarity between PCA transformation and VFT in (4) implies that eigenvalue λ_i encodes the variability of the dataset $\mathbf{X}_n$ in the direction of the principal component $\mathbf{v}_i$. In this context, the eigenvalues of the covariance matrix are the mathematical equivalent of the notion of graph frequencies in graph signal processing [45].

GNNs with convolutional filters that rely on a *linear shift-and-sum* operation fundamentally exhibit the stability to changes in graph topology [13]. Since the sample covariance matrix $\hat{\mathbf{C}}$ is likely to be perturbed with respect to $\mathbf{C}$ [49], stability is desirable to mitigate finite-sample effects on statistical inference. Motivated by this observation, we define the notion of coVariance filters that are polynomials in the covariance matrix and characterize the convolution operation in VNNs.

Definition 2 (coVariance Filters): Given a set of real valued, scalar parameters $\mathcal{H} = \{h_k\}_{k=0}^{K}$, the coVariance filter on a covariance matrix $\mathbf{C}$ is defined as

$$\mathbf{H}(\mathbf{C}) \triangleq \sum_{k=0}^{K} h_k \mathbf{C}^k. \tag{5}$$

Furthermore, the output of the covariance filter $\mathbf{H}(\mathbf{C})$ for an input $\mathbf{x}$ is given by

$$\mathbf{z} = \mathbf{H}(\mathbf{C})\mathbf{x}. \tag{6}$$

The application of coVariance filter $\mathbf{H}(\mathbf{C})$ on an input $\mathbf{x}$ translates to combining information across different sized neighborhoods. The spectral analysis of the covariance filtering operation in Definition 2 via VFT of the filter output $\mathbf{z}$ yields the frequency response of the covariance filter and reveals the similarities between covariance filtering and PCA. After taking the VFT of $\mathbf{z}$, we have

$$\tilde{\mathbf{z}} = \mathbf{V}^{\mathsf{T}}\mathbf{H}(\mathbf{C})\mathbf{x} = \sum_{k=0}^{K} h_k \mathbf{\Lambda}^k \mathbf{V}^{\mathsf{T}}\mathbf{x} = \sum_{k=0}^{K} h_k \mathbf{\Lambda}^k \tilde{\mathbf{x}}, \tag{7}$$

[1] GCNs and GNNs are used interchangeably in the rest of the paper.

[2] For ease of exposition, we henceforth use the notation $\mathbf{x}$ to refer to a realization of the random vector whose covariance matrix is $\mathbf{C}$.

where $\tilde{\mathbf{x}} = \mathbf{V}^\mathsf{T}\mathbf{x}$ is the covariance Fourier transform of $\mathbf{x}$ and (7) is valid due to the orthonormality of eigenvectors of $\mathbf{C}$. The frequency response of the coVariance filter depends on the filter taps $\mathcal{H} = \{h_k\}$ as well as the eigenvalues of $\mathbf{C}$, and is given by

$$h(\lambda) = \sum_{k=0}^{K} h_k \lambda^k. \tag{8}$$

Furthermore, since $\tilde{\mathbf{x}}$ is a projection of $\mathbf{x}$ on the eigenvector space $\mathbf{V}$ and $[\tilde{\mathbf{z}}]_i = h(\lambda_i)[\mathbf{V}^\mathsf{T}\mathbf{x}]_i$, the i-th element of $\tilde{\mathbf{z}}$ exhibits similarities with the standard PCA transformation. This observation is formalized in Lemma 1.

Lemma 1 (Spectrum of coVariance Filter and PCA): Given a covariance matrix $\mathbf{C}$ with eigendecomposition in (3), if the PCA transformation of input $\mathbf{x}$ is given by $\mathbf{q} = \mathbf{V}^\mathsf{T}\mathbf{x}$, there exists a filter bank of coVariance filters $\{\mathbf{H}_i(\mathbf{C}) : i \in \{1, \ldots, m\}\}$, such that, the score of the projection of input $\mathbf{x}$ on eigenvector $\mathbf{v}_i$ can be recovered by the application of a coVariance filter $\mathbf{H}_i(\mathbf{C})$ as:

$$[\mathbf{q}]_i = \mathbf{v}_i^\mathsf{T}\mathbf{H}_i(\mathbf{C})\mathbf{x}, \tag{9}$$

where the frequency response $h_i(\lambda)$ of the filter $\mathbf{H}_i(\mathbf{C})$ is given by

$$h_i(\lambda) = \begin{cases} 1, & \text{if} \quad \lambda = \lambda_i, \\ 0, & \text{otherwise} \end{cases}. \tag{10}$$

Lemma 1 asserts the equivalence between processing data samples with a weighted PCA transformation and with a specific polynomial on the covariance matrix.

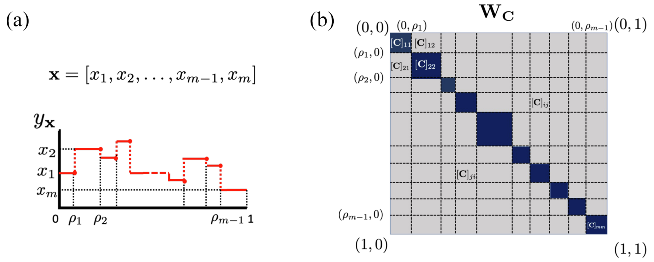
Our previous work in [5] showed that in contrast to PCA involving eigenvectors of the sample covariance matrix, information processing with polynomials of the sample covariance matrix can be stable to finite sample-induced perturbations. Indeed, in practice we may only have access to the sample covariance matrix $\hat{\mathbf{C}}$ which is an estimate of $\mathbf{C}$. Since $\hat{\mathbf{C}}$ is a consistent estimator of $\mathbf{C}$, the eigenvalues and eigenvectors of $\hat{\mathbf{C}}$ approach those of $\mathbf{C}$ in the limit of infinite number of samples, i.e., $n \to \infty$. However, for finite n, the eigenvectors and eigenvalues of $\hat{\mathbf{C}}$ are perturbed with respect to those of $\mathbf{C}$ [49]. In principle, statistical inference using PCA can be prone to instability due to eigenvectors corresponding to eigenvalues of the ensemble covariance matrix that are close [15] and, thus, lead to irreproducible statistical conclusions [16]. In this context, we informally state Theorem 2 from [5].

Lemma 2 (Stability of coVariance filter): Consider a dataset with sample covariance matrix $\hat{\mathbf{C}}$ formed by n samples and the counterpart ensemble covariance matrix $\mathbf{C}$. Under a mild assumption in [5], the following holds with high probability:

$$\left\|\mathbf{H}(\hat{\mathbf{C}}) - \mathbf{H}(\mathbf{C})\right\| = \mathcal{O}\left(\frac{\nu}{n^{\frac{1}{2}-\varepsilon}}\right), \tag{11}$$

for some $\nu > 0$ and $\varepsilon \in (0, 1/2)$.

Lemma 2 establishes that information processing using a polynomial of the covariance matrix offers stability with respect to the perturbations between the sample covariance matrix $\hat{\mathbf{C}}$ and $\mathbf{C}$ [5]. Also, as a corollary to Lemma 2, we can state that the difference between outputs of coVariance filters instantiated on distinct sample covariance matrices are bounded. These observations imply that statistical inference based on covariance filters is robust to finite sample size effects and, thus, result in consistent statistical outcomes with high confidence. No such guarantees are offered by approaches that leverage PCA-based transformation. Next, we discuss VNN architectures based on coVariance filters, which results in VNNs inheriting the stability offered by coVariance filters.



Representation of $\mathbf{x}$ on interval $[0, 1]$

Representation of $\mathbf{C}$ on space $[0, 1]^2$

Fig. 1. Representations of m-dimensional vector $\mathbf{x}$ and associated covariance matrix $\mathbf{C}$ in the continuous domain. (a) Representation for $\mathbf{x}$ is obtained by discretizing the interval [0,1]. (b) Representation $\mathbf{W_C}$ for $\mathbf{C}$ is obtained by discretizing the set $[0, 1]^2$ according to (18). Thus, $\mathbf{W_C}$ retains the symmetry of $\mathbf{C}$. The area spanned by the diagonal elements of $\mathbf{C}$ is marked in blue. The size of the square area allotted to a diagonal element is proportional to its value. Other parts of the grid accommodate the off-diagonal elements of $\mathbf{C}$.

B. Architecture

We begin with the description of a coVariance perceptron that forms one layer of the VNN architecture. To this end, we leverage a pointwise, nonlinear activation function $\sigma(\cdot)$, such that, for $\mathbf{x} = [x_1, \ldots, x_m]$, we have $\sigma(\mathbf{x}) = [\sigma(x_1), \ldots, \sigma(x_m)]$. Examples of pointwise, nonlinear activation functions are ReLU and tanh.

Definition 3 (coVariance Perceptron): For a given pointwise, nonlinear activation function $\sigma(\cdot)$, input $\mathbf{x}$, a coVariance filter $\mathbf{H}(\mathbf{C})$ and its corresponding set of filter taps $\mathcal{H}$, the coVariance perceptron is defined as

$$\Phi(\mathbf{x}; \mathbf{C}, \mathcal{H}) \triangleq \sigma(\mathbf{H}(\mathbf{C})\mathbf{x}). \tag{12}$$

A VNN can be constructed by stacking perceptrons to form a multi-layer information processing architecture. This observation is formalized next.

Remark 1 (Multi-layer VNN): Consider an L-layer architecture formed by stacking L coVariance perceptrons defined in Definition 3. In this scenario, we denote the coVariance filter in layer ℓ by $\mathbf{H}_\ell(\mathbf{C})$ and its corresponding set of filter taps are given by $\mathcal{H}_\ell$. For a given pointwise nonlinear activation function $\sigma(\cdot)$, the relationship between the input $\mathbf{x}_{\ell-1}$ and the output $\mathbf{x}_\ell$ for the coVariance perceptron in the ℓ-th layer is given by

$$\mathbf{x}_\ell = \sigma(\mathbf{H}_\ell(\mathbf{C})\mathbf{x}_{\ell-1}) \quad \text{for} \quad \ell \in \{1, \ldots, L\}, \tag{13}$$

where $\mathbf{x}_0$ is the input $\mathbf{x}$. We refer to this L-layer architecture as an L-layer VNN.

A figurative illustration of a multi-layer VNN is included in Fig. 1 in the Supplementary Material. Furthermore, sufficient expressive power can be accommodated in the VNN architecture via multiple input multiple output (MIMO) processing at every layer. Formally, consider a VNN layer ℓ that can process $F_{\ell-1}$ number of m-dimensional inputs and outputs F_ℓ number of m-dimensional outputs via $F_{\ell-1} \times F_\ell$ number of filter banks [13]. In this scenario, the input is specified as $\mathbf{X}_{\mathsf{in}} = [\mathbf{x}_{\mathsf{in}}[1], \ldots, \mathbf{x}_{\mathsf{in}}[F_{\mathsf{in}}]]$, and the output is specified as $\mathbf{X}_{\mathsf{out}} = [\mathbf{x}_{\mathsf{out}}[1], \ldots, \mathbf{x}_{\mathsf{out}}[F_{\mathsf{out}}]]$. The relationship between the f-th output $\mathbf{x}_{\mathsf{out}}[f]$ and the input $\mathbf{x}_{\mathsf{in}}$ is given by

$$\mathbf{x}_{\mathsf{out}}[f] = \sigma\left(\sum_{g=1}^{F_{\mathsf{in}}} \mathbf{H}_{fg}(\mathbf{C})\mathbf{x}_{\mathsf{in}}[g]\right), \tag{14}$$

where $\mathbf{H}_{fg}(\mathbf{C})$ is the coVariance filter that processes $\mathbf{x}_{\mathsf{in}}[g]$. Without loss of generality, we focus the subsequent discussion on the scenario when we have $F_\ell = F, \forall \ell \in \{1, \ldots, L\}$. In this case, the set of all filter taps is given by $\mathcal{H} = \{\mathcal{H}_{fg}^{\ell}\}, \forall f, g \in \{1, \ldots, F\}, \ell \in \{1, \ldots, L\}$, where $\mathcal{H}_{fg} = \{h_{fg}^{\ell}[k]\}_{k=0}^{K}$ and $h_{fg}^{\ell}[k]$ is the k-th filter tap for filter $\mathbf{H}_{fg}(\mathbf{C})$. Thus, we can compactly represent a multi-layer VNN architecture capable of MIMO processing via the notation $\Phi(\mathbf{x}; \mathbf{C}, \mathcal{H})$, as the set of filter taps $\mathcal{H}$ captures the full span of its architecture.

We also use the notation $\Phi(\mathbf{x}; \mathbf{C}, \mathcal{H})$ to denote the output of the VNN. For a VNN with F number of m-dimensional outputs in the final layer, the size of the VNN output $\Phi(\mathbf{x}; \mathbf{C}, \mathcal{H})$ is $m \times F$. The output $\Phi(\mathbf{x}; \mathbf{C}, \mathcal{H})$ is succeeded by a readout function that maps $\Phi(\mathbf{x}; \mathbf{C}, \mathcal{H})$ to the desired output. In this paper, we adopt a non-adaptive or non-learnable readout function (e.g., mean, max or min functions), which preserves the property of permutation invariance for the VNN model. Furthermore, a non-adaptive readout function is essential for the transferability property of VNNs (discussed in Section III).

It is prudent to study the robustness of VNN outputs to the number of samples n in order to guarantee reproducibility of VNN statistical outcomes. Specifically, it is desirable that the change in VNN outputs is controlled or bounded when the architecture is trained using sample covariance matrices estimated from n_1 or n_2 samples, when $n_1 \neq n_2$. In Theorem 1, we informally state the result on the stability of VNNs by analyzing $\|\Phi(\mathbf{x}; \hat{\mathbf{C}}, \mathcal{H}) - \Phi(\mathbf{x}; \mathbf{C}, \mathcal{H})\|$, i.e., the operator norm of the difference between the VNN outputs for the sample covariance matrix $\hat{\mathbf{C}}$ and the ensemble covariance matrix $\mathbf{C}$. This Theorem was also previously established in [5].

Theorem 1 (VNN Stability): Consider an ensemble covariance matrix $\mathbf{C}$ and its estimate $\hat{\mathbf{C}}$ formed from n samples. Given a bank of coVariance filters with filter taps $\mathcal{H} = \{\mathcal{H}_{fg}^{\ell} : f, g \in \{1, \ldots, F\}, \ell \in \{1, \ldots, L\}\}$, the coVariance filters are stable under a mild assumption in [5] and satisfy

$$\|\mathbf{H}_{fg}^{\ell}(\hat{\mathbf{C}}) - \mathbf{H}_{fg}^{\ell}(\mathbf{C})\| \leq \alpha_n, \tag{15}$$

for some $\alpha_n > 0$ with high probability (Lemma 2). Also, for a pointwise, nonlinear activation function $\sigma(\cdot)$, such that, $|\sigma(a) - \sigma(b)| \leq |a - b|$ for $a, b \in \mathbb{R}$, we have

$$\|\Phi(\mathbf{x}; \hat{\mathbf{C}}, \mathcal{H}) - \Phi(\mathbf{x}; \mathbf{C}, \mathcal{H})\| \leq LF^L \alpha_n. \tag{16}$$

The parameter α_n in (15) represents the finite sample effect on the perturbations in $\hat{\mathbf{C}}$ with respect to $\mathbf{C}$, and is borrowed from Lemma 2. By leveraging the perturbation theory of covariance matrices to analyze the stability of coVariance filters, we also show in the proof of Theorem 1 that α_n scales as $1/n^{\frac{1}{2}-\varepsilon}$, for some $\varepsilon \in (0, 1/2)$. We note that the bound in (16) becomes looser as F or L increases, which is consistent with the (less refined) results for GNNs [13]. However, without the analysis of lower bounds on $\|\Phi(\mathbf{x}; \hat{\mathbf{C}}, \mathcal{H}) - \Phi(\mathbf{x}; \mathbf{C}, \mathcal{H})\|$, we cannot claim that the robustness of VNNs indeed worsens with an increase in F or L. Moreover, we remark that VNNs sacrifice discriminability between eigenvectors associated with close eigenvalues to achieve stability [5]. As a corollary, we also state that Theorem 1 can readily be extended to characterize the difference between VNN outputs corresponding to sample covariance matrices estimated from n_1 and n_2 samples, via (16) and application of the triangle inequality.

III. Transferability of VNNs

The notion of transferability of VNNs across datasets of different dimensionalities is made feasible by the “scale-free” property of coVariance filters (Definition 2). From an implementation perspective, transferability of VNNs to a dataset of different number of features amounts to replacing the covariance matrix $\mathbf{C}$ in a VNN model $\Phi(\cdot; \mathbf{C}, \mathcal{H})$ with a covariance matrix of another size, while keeping the parameters $\mathcal{H}$ fixed. Since we consider covariance matrices of different dimensionalities, we denote a covariance matrix $\mathbf{C}$ of size $m \times m$ by $\mathbf{C}_m$. Informally, we can state our objective for assessing transferability as follows.

Informal problem statement for VNN transferability: Given a data point $\mathbf{x}_{m_1}$ from a dataset with m_1 features and associated covariance matrix $\mathbf{C}_{m_1}$, and another data point $\mathbf{x}_{m_2}$ from a dataset with m_2 features and associated covariance matrix $\mathbf{C}_{m_2}$, we aim to characterize the operator distance between VNN outputs $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$. If $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$ converge in some sense, we can conclude that the parameters $\mathcal{H}$ are transferable between two datasets consisting of m_1 and m_2 features.

A. Continuous Representation of a VNN

Note that the VNN outputs $\Phi(\cdot; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\cdot; \mathbf{C}_{m_2}, \mathcal{H})$ have distinct dimensionalities if $m_1 \neq m_2$ and therefore, a direct comparison between them is not natural. Fundamentally, it is imperative to provide a mathematical framework to compare vectors and covariance matrices of different sizes in order to facilitate transferability analyses of VNNs. To this end, we consider a simple mapping that represents the vector on a continuous interval [0,1].

Specifically, given an m-dimensional vector $\mathbf{x} = [x_1, \ldots, x_m]$, we can define a continuous representation of $\mathbf{x}$ as a function $y_{\mathbf{x}} : [0, 1] \mapsto \mathbb{R}$, such that, $y_{\mathbf{x}}(u) = x_i$ for $u \in \mathcal{U}_i$, where $\mathcal{U}_i$ is a pre-defined interval associated with the i-th element of $\mathbf{x}$. Similarly, we can map a covariance matrix $\mathbf{C}_m$ to a compact set $[0, 1]^2$ using the mapping $\mathbf{W}_{\mathbf{C}_m} : [0, 1]^2 \mapsto \mathbb{R}$, where we have $\mathbf{W}_{\mathbf{C}_m}(u, v) = [\mathbf{C}_m]_{ij}$ for $u \in \mathcal{U}_i$ and $v \in \mathcal{U}_j$. A pictorial illustration of $y_{\mathbf{x}}$ and $\mathbf{W}_{\mathbf{C}}$ for covariance matrix $\mathbf{C}$ is included in Fig. 1. Therein, the intervals $\mathcal{U}_i$ are parameterized by variables ρ_i, which will be discussed subsequently in (18). Note that we can recover $\mathbf{x}$ from $y_{\mathbf{x}}$ and vice-versa (similarly for $\mathbf{C}_m$ and $\mathbf{W}_{\mathbf{C}_m}$). Hence, for data points $\mathbf{x}_{m_1}$ and $\mathbf{x}_{m_2}$ consisting of m_1 and m_2 elements, respectively, the closeness of continuous representations $y_{\mathbf{x}_{m_1}}$ and $y_{\mathbf{x}_{m_2}}$ can be used as a metric to assess the similarity between data points in multi-scale datasets. This observation also extends to the comparison between matrices $\mathbf{C}_{m_1}$ and $\mathbf{C}_{m_2}$.

For a VNN architecture with F outputs in the final layer, the dimensionality of VNN outputs $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ is $m_1 \times F$ and for $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$ it is $m_2 \times F$. Thus, we can compare $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$ in terms of the continuous representations of every column in outputs $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$, where these continuous representations are defined in the same fashion as $y_{\mathbf{x}}$ above. For VNN $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$, we use the notation $y_{m_1}[f]$ to denote the continuous representation of the f-th output in $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$, i.e., $y_{[\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})]_f}$. Similar to the relationship between $y_{\mathbf{x}}$ and $\mathbf{x}$, the f-th VNN output $[\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})]_f$ and its continuous representation $y_{m_1}[f]$ are operationally interchangeable (see the Appendix for details). Using the continuous representations above, we can now formally describe the assessment of transferability of VNNs.

Formal problem statement for VNN transferability: Consider two VNNs $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$ instantiated on data with m_1 and m_2 features, respectively. If we have the following conditions: (a) the continuous approximations of inputs $\mathbf{x}_{m_1}$ and $\mathbf{x}_{m_2}$ are close, i.e., $\|y_{\mathbf{x}_{m_1}} - y_{\mathbf{x}_{m_2}}\|_2$ is bounded; and (b) the continuous approximations of covariance matrices $\mathbf{C}_{m_1}$ and $\mathbf{C}_{m_2}$ are close, i.e., $\|\mathbf{W}_{\mathbf{C}_{m_1}} - \mathbf{W}_{\mathbf{C}_{m_2}}\|_2$ is bounded; we aim to characterize the closeness between the continuous representations of VNN outputs $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$, i.e., find $\vartheta > 0$, such that,

$$\|y_{m_1}[f] - y_{m_2}[f]\|_2 \leq \vartheta, \forall f \in \{1, \ldots, F\}. \tag{17}$$

B. Mathematical Foundations of Transferability

The continuous representations of graph signals and graphs have previously been leveraged to study transferability of GNNs under the domain of graphon information processing [50]. Specifically, GNNs can be transferable between graphs belonging to a converging sequence if the graphs in this sequence converge to a limit object called *graphon*, as the number of nodes approaches infinity [51]. In a similar fashion, we leverage the theory of graphons [51] and graphon signal processing [50] to establish the transferability of VNNs. Graphons are the limits of *dense* graphs (i.e., graphs with number of edges of the order $\Theta(m^2)$) [52] and hence, appropriate to study limits of covariance matrices that are typically dense. The definition of a graphon is provided in Definition 4.

Definition 4 (Graphon): A graphon is a bounded, symmetric, measurable function $\mathbf{W} : [0, 1]^2 \mapsto [-1, 1]$.

To align the covariance matrix with the notion of graphon in Defintion 4, we can consider an appropriate scaling procedure (for instance, scaling the covariance matrix such that its maximum eigenvalue is 1). Recalling that a covariance matrix can be viewed as a weighted graph, a sequence of covariance matrices $\{\mathbf{C}_m\}$ being convergent implies that the sequence of their continuous representations, i.e., $\{\mathbf{W}_{\mathbf{C}_m}\}$, converges to some graphon $\mathbf{W}$ if $\mathbf{W}_{\mathbf{C}_m}$ is appropriately constructed from $\mathbf{C}_m$. This claim is based on generalizing [51, Corollary 3.9] to our setting and the formal statement is provided in Remark 2. This statement leverages the notion of *cut diatance* to characterize convergence. The definition of the cut distance between any two covariance matrices $\mathbf{C}_{m_1}$ and $\mathbf{C}_{m_2}$ is borrowed from the definition of cut distance between weighted graphs in [51, Sec. 2.3] and is also provided in the Supplementary Material.

Remark 2 (Graphon as limit object [51]): A sequence of covariance matrices $\{\mathbf{C}_m\}$ is deemed convergent if they form a Cauchy sequence with respect to the cut distance [51, Sec. 2.3]. Furthermore, for any convergent sequence of covariance matrices $\{\mathbf{C}_m\}$, the corresponding sequence of graphon approximations $\{\mathbf{W}_{\mathbf{C}_m}\}$ converges to a graphon $\mathbf{W}$.

To establish transferability of VNNs, we consider a converging sequence of covariance matrices $\{\mathbf{C}_m\}$ and investigate whether the parameters $\mathcal{H}$ can be transferred between any two VNNs instantiated on distinct covariance matrices in this sequence. The construction of $\mathbf{W}_{\mathbf{C}_m}$ relies on appropriately defining the intervals $\mathcal{U}_i$ and is described in the following steps [51, Sec. 3.1].

a) Partition the interval [0,1] into m disjoint intervals $[\mathcal{U}_1, \ldots, \mathcal{U}_m]$, such that,

$$\mathcal{U}_i = \begin{cases} [0, \rho_1] & \text{if} \quad i = 1 \\ (\rho_{i-1}, \rho_i] & \text{if} \quad i \in \{2, \ldots, m\} \end{cases}, \tag{18}$$

$$\text{where} \qquad \rho_i \triangleq \frac{1}{\mathsf{tr}(\mathbf{C}_m)} \sum_{j=1}^{i} [\mathbf{C}_m]_{jj}, \tag{19}$$

and $\mathsf{tr}(\mathbf{C}_m)$ is the trace of $\mathbf{C}_m$. Clearly, $\rho_m = 1$.

b) The relationship between feature i and feature j is given by $\mathbf{W}_{\mathbf{C}_m}(u_i, u_j) = [\mathbf{C}_m]_{ij}$ for $u_i \in \mathcal{U}_i, u_j \in \mathcal{U}_j$.

If the continuous representation $\mathbf{W}_{\mathbf{C}_m}$ of $\mathbf{C}_m$ is constructed according to the above steps, we refer to $\mathbf{W}_{\mathbf{C}_m}$ as the graphon approximation of $\mathbf{C}_m$. Thus, the graphon $\mathbf{W}$ forms the schema for which the covariance matrix $\mathbf{C}_m$ represents the covariance realization at resolution m. Our main result on the transferability of VNNs is contingent upon the following assumptions related to the covariance matrix $\mathbf{C}_m$, the graphon limit $\mathbf{W}$, and frequency response of the coVariance filters.

A1. ((Ω, ζ)-dominant property of covariance matrices.) For the sequence $\{\mathbf{C}_m\}$, there exist positive constants Ω and ζ, such that we have

$$\frac{1}{\mathsf{tr}(\mathbf{C}_m)} \max_{j \in \{1, \ldots, m\}} [\mathbf{C}_m]_{jj} \leq \frac{\Omega}{m^\zeta}, \tag{20}$$

for all finite m. We refer to the covariance matrix $\mathbf{C}_m$ satisfying the property in (20) as being (Ω, ζ)-dominant. This property implies that $\frac{1}{\mathsf{tr}(\mathbf{C}_m)} \max_{j \in \{1, \ldots, m\}} [\mathbf{C}_m]_{jj} \to 0$, as $m \to \infty$.

A2. (Lipschitz continuity of Graphon.) If $\mathbf{W}$ is the limit of the sequence $\{\mathbf{W}_{\mathbf{C}_m}\}$ as $m \to \infty$, then $\mathbf{W}$ satisfies

$$|\mathbf{W}(u_1, v_1) - \mathbf{W}(u_2, v_2)| \leq \alpha_1(|u_1 - u_2| + |v_1 - v_2|), \tag{21}$$

for any $u_1, v_1, u_2, v_2 \in [0, 1]$ and $\alpha_1 > 0$. Any graphon satisfying (21) is termed an α_1-Lipschitz graphon. The Lipschitz continuity of graphon $\mathbf{W}$ determines the smoothness of the information present between any two coordinates (u_1, v_1) and (u_2, v_2).

A3. The graphon approximation $\mathbf{W}_{\mathbf{C}_m}$ and the limit $\mathbf{W}$ satisfy $\mathbf{W}_{\mathbf{C}_m}(\rho_i, \rho_j) = \mathbf{W}(\rho_i, \rho_j)$, where ρ_i is defined in (19).

A4. (Lipschitz inputs to VNNs.) The continuous approximations of the inputs to VNNs are α_2-Lipschitz for some $\alpha_2 > 0$. Given an input $\mathbf{x}_{m_1}$, its continuous approximation $y_{\mathbf{x}_{m_1}}$ satisfies

$$|y_{\mathbf{x}_{m_1}}(u) - y_{\mathbf{x}_{m_1}}(v)| \leq \alpha_2 |u - v|, \quad \text{for } u, v \in [0, 1]. \tag{22}$$

A5.(Lipschitz coVariance filters.) The frequency response of a coVariance filter is α_3-Lipschitz for some $\alpha_3 > 0$, i.e., it satisfies $|h(\eta) - h(\hat{\eta})| \leq \alpha_3 |\eta - \hat{\eta}|$ for any pair of scalars $(\eta, \hat{\eta})$.

Assumption **A1** suggests that the variance profile of individual features in the dataset, characterized by their corresponding diagonal elements in the covariance matrix, must not be concentrated in a small subset of features. Also, since the covariance matrix $\mathbf{C}_m$ is considered to be a finite realization of graphon $\mathbf{W}$, a smaller Lipschitz constant α_1 in Assumption **A2** implies a smaller information mismatch between $\mathbf{W}$ and $\mathbf{W}_{\mathbf{C}_m}$ for a finite m, when $\mathbf{W}_{\mathbf{C}_m}$ and $\mathbf{W}$ satisfy the construction in Assumption **A3**. Assumption **A4** characterizes the smoothness of entries across the features of the dataset and, hence, the Lipschitz constant α_2 is a property of the given data. Assumption **A5** characterizes the variability in the coVariance filter outputs and is derived from the analyses of the transferability of VNNs.

Next, we state the main result of this section that establishes the transferability between VNNs processing datasets consisting of m_1 and m_2 features.

Theorem 2 (Transferability of VNNs): Consider two VNNs $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$ consisting of L layers and F outputs per layer. Under Assumptions A1-A5, the continuous representations of covariance matrices and inputs to the VNNs are close, i.e.,

$$\|\mathbf{W}_{\mathbf{C}_{m_1}} - \mathbf{W}_{\mathbf{C}_{m_2}}\|_2 \leq \alpha_1 \varrho(\Omega, \zeta, m_1, m_2), \tag{23}$$

$$\text{and} \quad \|y_{\mathbf{x}_{m_1}} - y_{\mathbf{x}_{m_2}}\|_2 \leq \alpha_2 \varrho(\Omega, \zeta, m_1, m_2). \tag{24}$$

Moreover, we have

$$\|y_{m_1}[f] - y_{m_2}[f]\|_2 \leq L F^L \alpha \varrho(\Omega, \zeta, m_1, m_2), \tag{25}$$

for $f \in \{1, \ldots, F\}$, where

$$\varrho(\Omega, \zeta, m_1, m_2) \triangleq \Omega^{3/2} \left(\frac{1}{m_1^{3\zeta/2 - 1}} + \frac{1}{m_2^{3\zeta/2 - 1}} \right), \tag{26}$$

$\alpha = (\alpha_1(\alpha_3 + \beta) + \alpha_2)$ for some contant $\beta > 0$, and $\zeta \in (2/3, 1]$.

Proof: See Appendix. ■

Theorem 2 implies that continuous representations of all F outputs of the respective final layers of VNNs $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$ converge as m_1 and m_2 grow. Since the continuous representation $y_{m_1}[f]$ and VNN output $[\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})]_f$ are operationally interchangeable, we expect the measures of central tendency (e.g., mean, median) of outputs $[\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})]_f$ and $[\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})]_f$ to converge as well. By extension, we also expect the measures of central tendency for $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$ to converge if Theorem 2 holds. In this context, if the VNN readout function is the unweighted mean, we expect the statistical outcomes of VNNs $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$ to be close and this convergence to be stronger for large m_1 and m_2. We also remark that Assumption A5 must hold *only* for a pair of scalars η and $\hat{\eta}$ that characterize the eigenvalues of the limiting graphon $\mathbf{W}$ and the graphon approximation $\mathbf{W}_{\mathbf{C}_m}$ in the proof of Theorem 2 and hence, is less stringent in practice.

The impact of Theorem 2 is broad, as we have shown that the parameters $\mathcal{H}$ can be "scale-free" while preserving the performance over an inference task. Specifically, a VNN can be instantiated on a dataset of different dimensionality than the training dataset and the VNN recovers close statistical outcomes for the same parameters $\mathcal{H}$ for both datasets, provided the data samples and covariance matrices of the training dataset and the new dataset are close in terms of their continuous representations. Thus, VNNs also offer a significant advantage over PCA-based analysis approaches as the principal components are restricted within the feature space of a dataset. They do not provide any mathematical insight into the structure of another dataset of different dimensionality, even when the datasets may be related. Fig. 2 provides an overview of the transferability of VNNs. Note that the theoretical results in Theorem 2 and the associated Assumptions **A1**-**A5** have been provided in the context of ensemble covariance matrices. In practice, the VNNs operate on sample covariance matrices, which are estimates of the ensemble covariance matrices. Under the inherent statistical uncertainty due to the finiteness of the data, Assumption **A3** may not be satisfied exactly for sample covariance matrices. Thus, the sample size and hence the closeness of the sample covariance matrices to the ensemble covariance matrices may also dictate the quality of transferability of VNNs.

As discussed previously, a multi-scale neuroimaging dataset provides an ideal setting for validating the transferability guarantees in Theorem 2. Indeed, VNNs also provide feature-level expressivity at the final layer. For instance, if a VNN is deployed for a regression task and the readout layer is a simple mean function, the VNN outputs can be used to characterize the contributions of each feature in the dataset to the regression outcomes. This can be of great value in neuroimaging applications, as each feature in a neuroimaging dataset is typically associated with a distinct brain region. Thus, VNNs naturally provide a feasible way to *interpret* the final statistical outcomes. Importantly, interpretability offered by VNNs can be traced to individual principal components of the covariance matrix. Hence, we contend VNNs are inherently interpretable, unlike other black-box deep learning architectures that rely on model-agnostic substrates for explainability, such as SHAP [53] or LIME [54]. Moreover, explanations offered by these methods may be unstable [38], [39], inconsistent [40], and computationally prohibitive [42]. Unlike a simpler statistical model such as PCA-based regression, VNNs offer theoretical stability guarantees that are of practical relevance; see [5] for an empirical demonstration of this

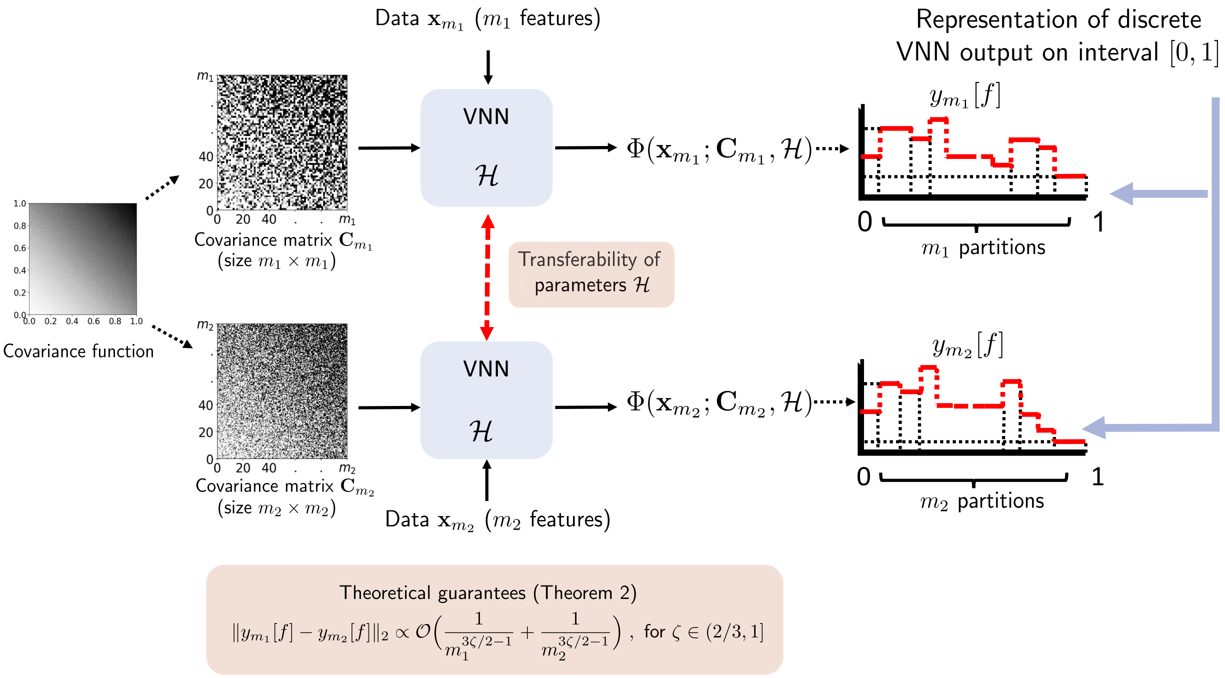

Data $\mathbf{x}_{m_1}$ (m_1 features)
Representation of discrete VNN output on interval $[0, 1]$
$y_{m_1}[f]$
VNN
$\mathcal{H}$
$\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$
0
1
m_1 partitions
Covariance matrix $\mathbf{C}_{m_1}$ (size $m_1 \times m_1$)
Covariance function
Transferability of parameters $\mathcal{H}$
$y_{m_2}[f]$
VNN
$\mathcal{H}$
$\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$
m_2 partitions
Covariance matrix $\mathbf{C}_{m_2}$ (size $m_2 \times m_2$)
Data $\mathbf{x}_{m_2}$ (m_2 features)
Theoretical guarantees (Theorem 2)
$\|y_{m_1}[f] - y_{m_2}[f]\|_2 \propto \mathcal{O}\left(\frac{1}{m_1^{3\zeta/2-1}} + \frac{1}{m_2^{3\zeta/2-1}}\right)$, for $\zeta \in (2/3, 1]$

Fig. 2. Overview of the transferability of VNNs.

advantage. We can further intertwine the interpretability with the transferability of VNNs to cross-validate findings across datasets of different dimensionalities. This desirable feature is lacking for most explainability methods in machine learning, which are primarily driven by sensitivity analyses of model outputs to perturbations in the input features. The observations made here motivate well investigating the utility of VNNs on multi-scale neuroimaging datasets; the subject dealt with next.

IV. Experiments

A. Multi-Scale FTDC Datasets

These datasets consist of the cortical thickness data extracted at different resolutions from a group of healthy controls (HC; $n = 105$, age $= 62.6 \pm 7.62$ years, 57 females) and a group of 67 individuals with mild cognitive impairment or Alzheimer's disease diagnosis (AD+; age $= 68.52 \pm 9.29$ years, 28 females). For each individual, the cortical thickness data was curated according to multi-resolution Schaefer atlas [22], at 100 parcel, 300 parcel, and 500 parcel resolutions with finer resolution cortical thickness estimates with increasing number of parcellations. The ANTs cortical thickness pipeline [55], [56] was used to derive mean cortical thickness within each atlas parcel using 3 T T1-weighted MRIs ($\sim$1 mm isotropic resolution). We investigate the transferability of VNNs on three datasets: FTDC100, FTDC300 and FTDC500, that constitute the cortical thickness datasets corresponding to 100, 300 and 500 cortical thickness features, respectively. Also, the FTDC100, FTDC300, and FTDC500 datasets are jointly referred to as FTDC datasets. All participants in the FTDC datasets took part in an informed consent procedure approved by an Institutional Review Board convened at University of Pennsylvania. The MRI data for FTDC datasets were provided by the Penn Frontotemporal Degeneration Center (NIH AG066597). Cortical thickness data were made available by Penn Image Computing and Science Lab at University of Pennsylvania. We remark that results consistent with the ones reported here have been obtained on publicly available datasets in recent work [57].

B. Transferability of VNNs for a Regression Task

VNNs were trained as regression models to predict chronological age using cortical thickness data of the HC group from FTDC datasets across different resolutions of Schaefer's atlas. To begin with, we remark that the sequence of covariance matrices formed by cortical thickness features extracted according to $100, 300, 500$ parcellations for HC group in FTDC datasets was converging (see Fig. 2 in the Supplementary Material, which uses the algorithm from [58] for implementation). This assessment was pertinent as our theoretical results in Theorem 2 hold for a converging sequence of covariance matrices. Our primary objective here is to show that predictive performance is transferable by VNN models across FTDC datasets, without re-training. Hence, this experiment is beyond the scope of traditional multivariate regression approaches that rely on a PCA-based transformation in the first step, followed by a regression model.

VNN learning: We trained three sets of VNN models; one each for the HC group in FTDC100, FTDC300, and FTDC500 datasets. The training process was similar for all VNNs. Note that the VNN output of the architecture represented by $\Phi(\mathbf{x}; \hat{\mathbf{C}}, \mathcal{H})$[3] for one m-dimensional input is of dimension

[3] We use the notation $\hat{\mathbf{C}}$ for covariance matrix here, as the VNN architecture is instantiated on the sample covariance matrix in practical implementations.

$m \times F$ if the VNN architecture has F m-dimensional outputs in the final layer. The regression output is determined by a readout layer which evaluates an unweighted mean of all outputs of the final layer of VNN. Therefore, the regression output for a vector of cortical thickness features $\mathbf{x}$ is given by

$$\hat{y} = \frac{1}{Fm}\sum_{j=1}^{m}\sum_{f=1}^{F}[\Phi(\mathbf{x};\hat{\mathbf{C}},\mathcal{H})]_{jf}. \tag{27}$$

Prediction using unweighted mean at the output implies that the VNN model exhibits permutation-invariance (i.e., the final output is independent of the permutation of the input features and covariance matrix) and transferability. For a regression task, the training dataset $\{\mathbf{x}_i, y_i\}_{i=1}^{n}$ is used to learn the filter taps in $\mathcal{H}$ for the VNN $\Phi(\cdot;\hat{\mathbf{C}},\mathcal{H})$ such that they minimize the training loss, i.e.,

$$\mathcal{H}_{\mathsf{opt}} = \min_{\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}\ell(\hat{y}_i, y_i), \tag{28}$$

where $\hat{y}_i$ is evaluated similar to (27) and $\ell(\cdot)$ is the mean-squared error (MSE) loss function.

Each dataset was split into an approximately $90/10$ train/test split. Thus, the test sets for FTDC datasets consisted of 10 individuals. The sample covariance matrix was evaluated using all samples in the training set ($n = 95$) and we had the sample covariance matrix $\hat{\mathbf{C}}$ of size $m \times m$ (where $m = 100$ for FTDC100, $m = 300$ for FTDC300, $m = 500$ for FTDC500). Furthermore, for all datasets, $\hat{\mathbf{C}}$ was normalized such that its maximum eigenvalue was 1. Next, the training set was randomly split internally, such that, the VNN was trained with respect to the MSE loss between the predicted age and the true age in $n = 84$ samples for FTDC datasets. The loss was optimized using batch stochastic gradient descent with Adam optimizer available in PyTorch library [59] for up to 100 epochs. The batch size was 34 for FTDC100 dataset, 8 for FTDC300 dataset, and 12 for FTDC500 dataset. The VNN model with the minimum MSE on the remaining samples in the training set (which acted as a validation set) was included in the set of nominal models for this permutation of the training set. For each dataset, we trained and validated the VNN models over 100 permutations of the complete training set of $n = 95$ samples for each of the FTDC datasets, thus, leading to 100 trained VNN models (also referred to as nominal models) per dataset.

The hyperparameters for the VNN architecture and learning rate of the optimizer were chosen according to a hyperparameter search procedure using the package Optuna [60]. For FTDC100, the VNN had a $L = 2$-layer architecture, with a filter bank such that we had $F = 26$ and 2 filter taps in each layer. The learning rate for the optimizer was 0.058. The number of learnable parameters for this VNN was 1456. For FTDC300, the VNN had a $L = 2$-layer architecture, with a filter bank such that we had $F = 39$ and 3 filter taps in the first layer and 2 filter taps in the second layer. The learning rate for the optimizer was 0.0241. The number of learnable parameters for this VNN was 3237. For FTDC500, the VNN model had a $L = 2$-layers with a filter bank such that we had $F = 27$ and 4 filter taps in the first layer and 2 filter taps in the second layer. The number of learnable parameters for this VNN was 1620. The learning rate for the Adam optimizer was set to 0.0631.

TABLE I
TRANSFERABILITY ACROSS DATASETS (MAE FOR VNN REGRESSION OUTPUTS WITH RESPECT TO THE GROUND TRUTH)

Testing Training	FTDC100 (HC)	FTDC300 (HC)	FTDC500 (HC)
FTDC100 (HC)	5.39 ± 0.084	5.5 ± 0.101	5.61 ± 0.132
FTDC300 (HC)	5.39 ± 0.193	5.41 ± 0.167	5.47 ± 0.169
FTDC500 (HC)	5.43 ± 0.2	5.38 ± 0.15	5.4 ± 0.169

TABLE II
TRANSFERABILITY ACROSS DATASETS (PEARSON'S CORRELATION BETWEEN VNN OUTPUTS AND GROUND TRUTH)

Testing Training	FTDC100 (HC)	FTDC300 (HC)	FTDC500 (HC)
FTDC100 (HC)	0.49 ± 0.017	0.47 ± 0.018	0.468 ± 0.018
FTDC300 (HC)	0.498 ± 0.05	0.49 ± 0.042	0.486 ± 0.04
FTDC500 (HC)	0.51 ± 0.021	0.509 ± 0.02	0.51 ± 0.021

Since the readout layer in all trained VNNs was non-adaptive and it evaluated the unweighted mean of the outputs of the final VNN layer to form an estimate for chronological age, the trained VNN could readily process a dataset with different dimensionality without any retraining or alteration to the architecture. The performance outcomes were quantified in terms of mean absolute error (MAE) and Pearson's correlation between the VNN output and ground truth.

Results: We tabulate MAE in Table I and Pearson's correlation between ground truth and VNN output in Table II. Since the objective is to illustrate transferability of VNNs over different scales, the MAE and Pearson's correlation results are reported for complete datasets. These metrics for only the test sets have been provided in the Supplementary Material. For both tables, the row ID provides the dataset on which VNN models were trained and the column ID indicates the dataset for which the VNN performance is reported (after transferring the VNNs if training and testing datasets are different). For instance, the element with row ID "FTDC100" and column ID "FTDC300" in Table I represents the mean and standard deviation of MAE evaluated on FTDC300 dataset ($m = 300$) for the 100 nominal VNN models trained on FTDC100 dataset ($m = 100$). The elements with same row ID and column ID in Tables I and II provide the baseline performance to gauge performance after transferring VNNs.

The results in Tables I and II show that the performance of VNNs in terms of MAE and correlation between VNN output and ground truth was preserved after transferring VNNs across FTDC datasets that were curated according to different resolutions of Schaefer's atlas. The transferability of VNNs across FTDC datasets was corroborated by scatterplots (see Fig. 4 in the Supplementary Material). We also remark that this experiment is not feasible for PCA-regression models as the principal components and the regression model from one dataset cannot be naively transferred to process another dataset that has a different number of features. Next, we demonstrate

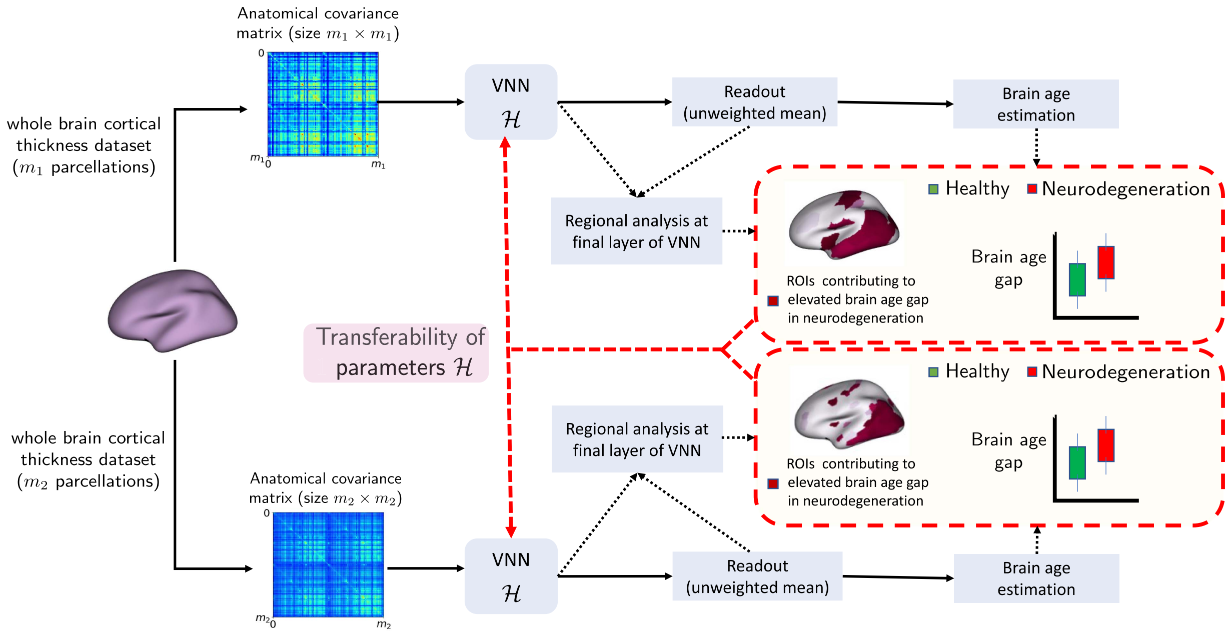

Anatomical covariance matrix (size $m_1 \times m_1$)
0
m_1 0
m_1
whole brain cortical thickness dataset (m_1 parcellations)
VNN
$\mathcal{H}$
Readout (unweighted mean)
Brain age estimation
Regional analysis at final layer of VNN
Healthy
Neurodegeneration
Brain age gap
ROIs contributing to elevated brain age gap in neurodegeneration
Transferability of parameters $\mathcal{H}$
whole brain cortical thickness dataset (m_2 parcellations)
Anatomical covariance matrix (size $m_2 \times m_2$)
0
m_2 0
m_2
VNN
$\mathcal{H}$
Regional analysis at final layer of VNN
Healthy
Neurodegeneration
Brain age gap
ROIs contributing to elevated brain age gap in neurodegeneration
Readout (unweighted mean)
Brain age estimation

Fig. 3. Cross-validation of Δ-Age and associated anatomical interpretability by leveraging the transferability of VNNs. If the transferability of parameters $\mathcal{H}$ for the chronological age prediction task holds for different datasets of m_1 and m_2 features (according to Theorem 2), we expect to see similar Δ-Age distributions and associated anatomical interpretability (characterized by regions of interest (ROIs) highlighted in red color on the brain templates) across them without re-training.

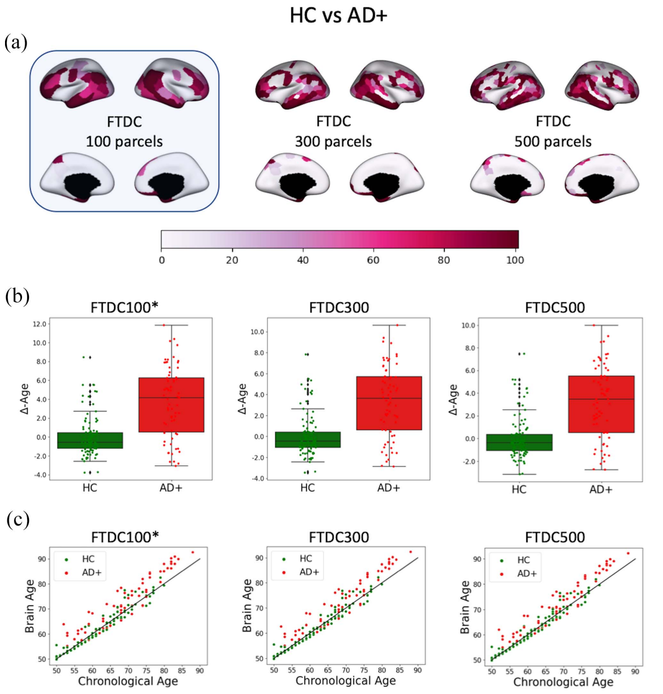


Fig. 4. Brain age prediction across datasets curated according to multiple scales of Schaefer's atlas. Panel (a) illustrates the regional profiles consisting of brain regions with robust, elevated regional residuals in the combined AD+ group with respect to the HC group. The VNNs were trained to predict chronological age on FTDC100 and the robustness was evaluated over 100 nominal VNN models. The regional profiles were obtained for the datasets with 300 features and 500 features after transferring the VNNs from FTDC100 to FTDC300 and FTDC500. Panel (b) displays the box plots for Δ-Age corresponding to the regional profiles in Panel (a). Panel (c) plots brain age versus chronological age for datasets with 100, 300, and 500 cortical thickness features.

the utility of regression models trained in Section IV-B in the task of anatomically interpretable brain age prediction. Besides sacrificing the transferability property of VNNs, the usage of adaptive readout functions can also potentially impact the interpretability offered by VNNs in the context of brain age prediction (Appendix I in [28]). Hence, VNNs trained as regression models without adaptive readout functions can provide an explainable perspective to an inference task, albeit without achieving the best possible performance.

C. Transferability of Interpretability in Δ-Age Prediction Task

Δ-Age is a known biomarker of cognitive decline and neurodegeneration [29], [33]. Also, age is a major risk factor for Alzheimer's disease (AD) and hence, AD is characterized by biological traits that signify accelerated aging [61]. We start by leveraging our Δ-Age prediction pipeline from [28] to provide an anatomical perspective to Δ-Age in AD in FTDC100 dataset.

1) Δ-Age Prediction in FTDC100: A layman overview of the procedure of inferring Δ-Age using neuroimaging data can be summarized in three steps.

- *Step 1:* Train a VNN model to predict chronological age of a healthy population from a neuroimaging dataset.
- *Step 2:* If the correlation between chronological age estimate and ground truth is smaller than 1, it may induce an age-related bias in the VNN model output (implying underestimation for older individuals and overestimation for younger individuals). Hence, an age-bias correction model (e.g., using linear regression) is applied to correct for this bias in the VNN model outputs.
- *Step 3:* The output of the VNN model after age-bias correction forms the brain age estimate. The difference between

the brain age estimate and chronological age provides the Δ-Age for an individual.

In Step 1, we utilize the VNN models trained to predict chronological age for FTDC100 group in Section IV-B. Because the regression output by the VNN model was determined by unweighted aggregation of the final layer outputs, it can be conceptualized as an unweighted mean of age predictions at individual brain regions. Therefore, the VNN architecture allows us to compute "regional residuals" (scalar output at a given region derived from VNN final layer output - aggregated VNN output or age estimate formed by VNN) at each brain region to assess their contribution to the final output of VNN. This procedure is formalized next.

Identification of regions associated with neurodegeneration: The VNN architecture allows us to associate a scalar output with each of the m dimensions in the final layer. Specifically, we have

$$\mathbf{p}_i = \frac{1}{F}\sum_{f=1}^{F}[\Phi(\mathbf{x}_i; \hat{\mathbf{C}}_m, \mathcal{H})]_f, \tag{29}$$

where $\mathbf{p}_i$ is the vector denoting the mean of all final layer outputs associated with filters in the filter bank at the final layer. Note that the mean of all elements of $\mathbf{p}_i$ is the prediction $\hat{y}_i$ formed in (27). In the context of cortical thickness datasets, we can associate each element of $\mathbf{p}_i$ with a distinct brain region. Therefore, the vector $\mathbf{p}_i$ is a vector of "regional contributions" to the output $\hat{y}_i$ by the VNN. The parameters $\mathcal{H}$ were learnt over the HC group as described previously and kept unchanged in the subsequent analyses. We use the notation $\hat{\mathbf{C}}_{100}$ for the covariance matrix formed by the cortical thickness features from HC group.

Next, we leverage (29) to capture the effect of neurodegeneration on brain regions. For this purpose, in the FTDC100 dataset, we evaluated the covariance matrix $\hat{\mathbf{C}}_{100}^{\text{AD+}}$ from the combined cortical thickness data of HC and AD+ groups. Note that the VNN models were not re-trained for Δ-Age evaluations and hence, were oblivious to the AD+ group during training. For every individual in the combined dataset of HC and AD+ groups, we processed their cortical thickness data $\mathbf{x}$ through the VNN model $\Phi(\mathbf{x}; \hat{\mathbf{C}}_{100}^{\text{AD+}}, \mathcal{H})$, where parameters $\mathcal{H}$ were learnt in the regression task on the data from HC group as described previously. Hence, the mean vector of all final layer outputs for cortical thickness input $\mathbf{x}$ is given by

$$\mathbf{p} = \frac{1}{F}\sum_{f=1}^{F}[\Phi(\mathbf{x}; \hat{\mathbf{C}}_{100}^{\text{AD+}}, \mathcal{H})]_f, \tag{30}$$

and the VNN output is $\hat{y} = \frac{1}{100}\sum_{j=1}^{100}[\mathbf{p}]_j$. Furthermore, we define the vector of residuals as $\mathbf{r}$, whose a-th element (associated with brain region represented by feature a in this case) is given by

$$[\mathbf{r}]_a \triangleq [\mathbf{p}]_a - \hat{y}. \tag{31}$$

Thus, (31) allows us to characterize the residuals with respect to the VNN output $\hat{y}$ at the regional level for an individual with cortical thickness data $\mathbf{x}$. Henceforth, we refer to the residuals evaluated according to (31) as "regional residuals". We hypothesized that a larger brain age in a neurodegenerative condition could be linked to an aggregated effect of contributions from certain biologically plausible brain regions.

The population of residual vectors for HC group is denoted by $\mathbf{r}_{\text{HC}}$ and that for individuals in AD+ group by $\mathbf{r}_{\text{AD+}}$. ANOVA was performed to test for group differences between the regional residuals from the individuals in HC and AD+ groups. Also, since the objective is to capture accelerated aging, our results focus only on elevated regional residual distribution in AD+ group with respect to HC group. Further, the group difference between AD+ and HC groups in the residual vector element for a brain region was deemed significant if it met the following criteria: i) the corrected p-value (Bonferroni correction) for the clinical diagnosis label in the ANOVA model was smaller than 0.05; and ii) the uncorrected p-value for clinical diagnosis label in ANCOVA model with age and sex as covariates was smaller than 0.05. An example for this regional analysis of VNN outputs is included in [28, Appendix F]. The analysis of regional residuals described above was performed for each trained VNN model, and we tabulated the number of VNN models for which a brain region was deemed to have a higher regional residual in the AD+ group with respect to the HC group. A higher number of VNN models isolating a brain region as significant suggested higher robustness of the effect observed for that brain region. The `fsbrain` package in R was used to project the robustness of significantly elevated regional residuals for a brain region on the brain template [62].

Subject-level brain age prediction: In general, the systemic bias in the gap between $\hat{y}$ and y, where the age may be underestimated for older individuals and overestimated for younger individuals, may confound the interpretations of brain age [63]. Therefore, to correct for this age-driven bias, we adopted a linear regression model based approach to correct any age bias in the VNN age estimates [63], [64]. Specifically, the VNN estimate $\hat{y}$ was bias-corrected to obtain the brain age $\hat{y}_{\text{B}}$ for an individual with chronological age y and cortical thickness data $\mathbf{x}$, by adopting the following procedure.

- *Step 1:* Fit a linear regression model on the complete dataset to determine coefficients α and β in the following linear model: $\hat{y} - y = \alpha y + \beta$.
- *Step 2:* Evaluate brain age as follows: $\hat{y}_{\text{B}} = \hat{y} - (\alpha y + \beta)$.

For an individual with cortical thickness $\mathbf{x}$ and chronological age y, the brain age gap Δ-Age is defined as

$$\Delta\text{-Age} \triangleq \hat{y}_{\text{B}} - y. \tag{32}$$

The linear regression model in the age-bias correction procedure was trained only for the HC group to account for bias in the VNN estimates due to healthy aging, and then applied to the AD+ group. Further, the distributions of Δ-Age were obtained for all individuals in HC and AD+ groups. We verified that differences in Δ-Age for AD+ and HC group were not driven by age or gender differences via ANCOVA with age and sex as covariates.

Results: The brain regions with significantly elevated regional residuals in AD+ with respect to HC are projected on the brain template for the FTDC100 dataset in Fig. 4(a). Specifically, Fig. 4(a) displays the robustness (from analyses of 100 VNN models) for various brain regions in having an elevated regional effect in their corresponding residual elements for AD+ group with respect to HC group. The highlighted brain regions were concentrated in various biologically plausible regions for AD, such as, bilateral medial temporal, temporal pole, entorhinal, and frontal regions. Additional experiments showed that the anatomical interpretability inferred by the analyses of residual elements was highly correlated with specific eigenvectors of $\hat{\mathbf{C}}_{100}^{\mathrm{AD+}}$ (see Supplementary Material).

The Δ-Age for AD+ group was 3.67 ± 3.73 years and for HC was 0 ± 2.06 years. Hence, as expected, the Δ-Age tended to be elevated for the AD+ group. Fig. 4(b) displays the box plots for Δ-Age in HC and AD+ groups for FTDC100 dataset and Fig. 4(c) displays the scatter plots between brain age and chronological age for both groups.

2) Cross-Validation on FTDC300 and FTDC500 Datasets Using Transferability of VNNs: Next, we leverage the transferability of VNNs to cross-validate the Δ-Age results obtained via analyses of FTDC100 dataset on FTDC300 and FTDC500 datasets. For this purpose, the VNNs trained on FTDC100 dataset were transferred to predict chronological age in FTDC300 and FTDC500 datasets, followed by age-bias correction to obtain brain age and Δ-Age in both scenarios. Our approach to cross-validating Δ-Age and its associated anatomical interpretability is depicted in Fig. 3.

Due to the transferability of VNNs across FTDC datasets, Δ-Age profiles and brain age versus chronological age plots for FTDC300 and FTDC500 datasets in Fig. 4 were observed to be consistent with that for FTDC100 dataset. Hence, the transferability of VNNs enabled us to recover results similar to that of FTDC100 dataset in FTDC300 and FTDC500 datasets without re-training. This observation suggests that Δ-Age inferred by VNNs was transferable, and its anatomical interpretability was robust across different parcellations of Schaefer's atlas.

V. Conclusion

The graph convolution operator on a covariance matrix, termed as a coVariance filter, forms the backbone of the VNN architecture. The coefficients of the coVariance filter characterize its ability to manipulate the data according to the eigenspectrum of the covariance matrix to achieve a learning objective. Thus, statistical inference using VNNs draws similarities with PCA-driven statistical approaches. However, PCA conventionally operates within the feature space of a given dataset and hence, does not provide any notion of similarity between principal components of datasets with different number of features. In this paper, we have studied the key property of transferability of VNN models, which allows VNNs to be transferable between datasets with similar characteristics but different number of features. The notion of similarity between datasets consisting of different number of features is borrowed from the existing theory of graphons that studies limits of dense graphs [52]. Specifically, our theoretical results have shown that if there exists a sequence of covariance matrices that converges to a continuous limit object in the limit of infinite number of features, then VNNs can be transferred between any two covariance matrices of such a sequence for statistical inference. The underlying theoretical results rely on the convergence of the eigenspectrum of a continuous approximation of covariance matrices, which result in convergence of the coVariance filter outputs for covariance matrices belonging to a converging sequence, and subsequently, the convergence of VNN outputs. Our experiments pertain to dense anatomical covariance matrices and therefore, graphon model-based analyses were certainly appropriate to study transferability of VNNs. Furthermore, sparse covariance matrices are also of practical interest as they can help manage computational complexity [65]. Therefore, studying VNN transferability over sparse covariance matrices is a future direction of interest.

In the experiments in Section IV-C, VNNs that were trained on the healthy population were deployed on a population with AD diagnosis. Thus, in principle, VNNs learned information about healthy aging from the healthy population and were able to quantify accelerated aging as a biomarker in AD. Furthermore, the transferability of VNNs to datasets of various dimensionalities and populations in different clinical contexts draws similarities with the adaptability and transference of large-scale foundation models [66]. The observations made here could further be extended to study VNNs trained on healthy population as foundation models for biomarkers for various health conditions in future work [57].

Appendix
Graphon Information Processing

The theory of graphons has previously been leveraged to study the transferability of GNNs between graphs in the same graphon family [14]. The proof of Theorem 2 relies on establishing the transferability of VNNs between datasets in the setting where their corresponding covariance matrices belong to a converging sequence characterized by a graphon. Our first objective in this section is to show that data processing over coVariance filter can equivalently be represented in the continuous domain using its graphon approximation. Establishing this property will ultimately allow us to compare VNNs instantiated on covariance matrices derived from datasets with different numbers of features. Using the theory of convergence of graphons and interpreting the covariance matrix as a weighted graph representation of data, a graphon $\mathbf{W}$ exists as a limiting object for the sequence of graphon approximations $\{\mathbf{W}_{\mathbf{C}_m}\}$ if the sequence of covariance matrices $\{\mathbf{C}_m\}$ converges in the *cut distance* [51]. A distinct feature of the cut distance is that it allows the comparison of covariance matrices of different sizes. Hence, all covariance matrices whose graphon approximations converge to a graphon can be considered to be a part of that graphon family.

A Information Processing With Graphons

We next show that a coVariance filter $\mathbf{H}(\mathbf{C}_m)$ can be equivalently represented in the continuous domain using convolution operations over graphon representations $\mathbf{W}_{\mathbf{C}_m}$. Given a coVariance filter output $\mathbf{z} = \mathbf{H}(\mathbf{C}_m)\mathbf{x}$, the continuous representation of $\mathbf{x}$ is $y_{\mathbf{x}}$ and that of $\mathbf{C}_m$ is $\mathbf{W}_{\mathbf{C}_m}$. The operation $\mathbf{C}\mathbf{x}$ is fundamental to the convolution operation in $\mathbf{H}(\mathbf{C}_m)\mathbf{x}$ and therefore, we first provide its continuous equivalent. For $\mathbf{s} = \mathbf{C}\mathbf{x}$, the i-th element of $\mathbf{s}$ is

$$[\mathbf{s}]_i = \sum_{j=0}^{m} [\mathbf{C}_m]_{ij} [\mathbf{x}]_j. \tag{33}$$

Thus, $[\mathbf{s}]_i$ is a linear combination of elements in $\mathbf{x}$ according to the i-th row of $\mathbf{C}_m$. In the continuous space, we can equivalently write (33) as

$$y_{\mathbf{s}}(u) = \int_0^1 \mathbf{W}_{\mathbf{C}_m}(u,v) y_{\mathbf{x}}(v) \mathrm{d}v, \tag{34}$$

where $y_{\mathbf{x}}$ is the continuous representation of $\mathbf{x}$ obtained according to the intervals defined in (18). Note that $y_{\mathbf{s}}$ is a continuous representation of $\mathbf{s}$, i.e., they satisfy $y_{\mathbf{s}}(u) = [\mathbf{s}]$ for $u \in \mathcal{U}_i$. Hence, $y_{\mathbf{s}}$ and $\mathbf{s}$ can be recovered from each other. This observation can be extrapolated to define the continuous equivalent of a coVariance filter. This is feasible because we can write the entity $\mathbf{C}_m^k \mathbf{x}$ in $\mathbf{H}(\mathbf{C})$ in a recursive form. Specifically, if we have $\mathbf{s}_k = \mathbf{C}_m^k \mathbf{x}$, then we can rewrite $\mathbf{s}_k$ as $\mathbf{s}_k = \mathbf{C}_m \mathbf{s}_{k-1}$, where $\mathbf{s}_0 = \mathbf{x}$. Thus, using the same reasoning that established the equivalence between (33) and (34), we conclude that the continuous representation $y_{\mathbf{s}_k}$ of $\mathbf{s}_k$ can be recovered via the following operation

$$y_{\mathbf{s}_k}(u) = \int_0^1 \mathbf{W}_{\mathbf{C}_m}(u,v) y_{\mathbf{s}_{k-1}}(v) \mathrm{d}v. \tag{35}$$

Since the coVariance filter output $\mathbf{z}$ is a weighted aggregation of the terms $\mathbf{s}_k$, we can write its continuous representation $y_{\mathbf{z}}$ as

$$y_{\mathbf{z}}(u) = \sum_{k=0}^{K} h_k y_{\mathbf{s}_k}(u). \tag{36}$$

Using the mathematical steps leading up to (36), we have shown that the continuous representation of the covariance filter output $\mathbf{z}$ can be recovered via the convolution operations over the graphon representation $\mathbf{W}_{\mathbf{C}_m}$ in (34) and (35). Also, $\mathbf{z}$ and $y_{\mathbf{z}}$ are operationally interchangeable. Moreover, we can also extrapolate this correspondence between $\mathbf{z}$ and $y_{\mathbf{z}}$ to covariance perceptrons and VNNs with multi-layer architecture and MIMO information processing. The extension of this observation to coVariance perceptron and a basic VNN is trivial as the coVariance output is evaluated after application of pointwise non-linearity σ on $\mathbf{z}$ and a basic VNN is formed by stacking multiple coVariance perceptrons and number of inputs and outputs at each layer (i.e., F) being set to 1.

We use the notation $\mathbf{x}_m$ to denote an input vector with m features. Thus, if VNN output $\Phi(\mathbf{x}_m; \mathbf{C}_m, \mathcal{H})$ is of size $m \times 1$ and we have $F = 1$ and number of layers L, its continuous approximation $y_{\Phi(\mathbf{x}_m; \mathbf{C}_m, \mathcal{H})}$ can be recovered by a convolutional architecture instantiated on $\mathbf{W}_{\mathbf{C}_m}$ with input $y_{\mathbf{x}_m}$. For a VNN with MIMO processing, each VNN layer has multiple m-dimensional inputs and multiple m-dimensional outputs. Thus, we can equivalently define an architecture capable of performing MIMO processing that is instantiated on $\mathbf{W}_{\mathbf{C}_m}$ and $\mathbf{x}_m$ and produces multiple continuous representations as the output. Such an architecture has previously been studied in the form of graphon neural networks [50]. In this context, we define the model $\tilde{\Phi}(y_{\mathbf{x}_m}; \mathbf{W}_{\mathbf{C}_m}, \mathcal{H})$ that is modeled via convolution operations over $\mathbf{W}_{\mathbf{C}_m}$ in (35) and has the same architecture as the VNN $\Phi(\mathbf{x}_m; \mathbf{C}_m, \mathcal{H})$. Note that the outputs of $\tilde{\Phi}(y_{\mathbf{x}_m}; \mathbf{W}_{\mathbf{C}_m}, \mathcal{H})$ are continuous representations of the outputs of VNN $\Phi(\mathbf{x}_m; \mathbf{C}_m, \mathcal{H})$ (see also Fig. 2 for an illustration). Thus, we can investigate the transferability of parameters $\mathcal{H}$ between VNNs instantiated on covariance matrices $\mathbf{C}_{m_1}$ and $\mathbf{C}_{m_2}$ by analyzing the difference between $\tilde{\Phi}(y_{\mathbf{x}_{m_1}}; \mathbf{W}_{\mathbf{C}_{m_1}}, \mathcal{H})$ and $\tilde{\Phi}(y_{\mathbf{x}_{m_2}}; \mathbf{W}_{\mathbf{C}_{m_2}}, \mathcal{H})$.

In this context, our analysis hinges on the setting in which the graphon approximations $\mathbf{W}_{\mathbf{C}_{m_1}}$ and $\mathbf{W}_{\mathbf{C}_{m_2}}$ belong to a sequence of graphon approximations $\{\mathbf{W}_{\mathbf{C}_m}\}$ that converges to a graphon $\mathbf{W}$. Thus, we also consider an information processing architecture $\tilde{\Phi}(y; \mathbf{W}, \mathcal{H})$ instantiated on graphon $\mathbf{W}$, such that y and continuous representations $y_{\mathbf{x}_m}$ always satisfy $y_{\mathbf{x}_m}(\rho_i) = y(\rho_i), \forall i \in \{1, \ldots, m\}$. Here, we can also interpret $\tilde{\Phi}(y; \mathbf{W}, \mathcal{H})$ as a generative model with $\tilde{\Phi}(y_{\mathbf{x}_m}; \mathbf{W}_{\mathbf{C}_m}, \mathcal{H})$ being an instance of $\tilde{\Phi}(y; \mathbf{W}, \mathcal{H})$ at resolution m. Thus, our analysis of transferability of VNNs also includes the study of convergence of outputs from $\tilde{\Phi}(y_{\mathbf{x}_m}; \mathbf{W}_{\mathbf{C}_m}, \mathcal{H})$ with that from $\tilde{\Phi}(y; \mathbf{W}, \mathcal{H})$.

To this end, we now formally define a convolution filter over a graphon and characterize its frequency response. We denote the k-hop aggregation (analogous to $\mathbf{C}^k \mathbf{x}$) on $\mathbf{W}_{\mathbf{C}_m}$ and continuous representation $y_{\mathbf{x}_m}$ by the operator $T^k_{\mathbf{W}_{\mathbf{C}_m}} y_{\mathbf{x}_m}$ that is given by

$$(T^k_{\mathbf{W}_{\mathbf{C}_m}} y_{\mathbf{x}_m})(u) \triangleq \int_0^1 \mathbf{W}_{\mathbf{C}_m}(u,v)(T^{k-1}_{\mathbf{W}_{\mathbf{C}_m}} y_{\mathbf{x}_m})(v) \mathrm{d}v, \tag{37}$$

for any $k > 1$, where

$$(T_{\mathbf{W}_{\mathbf{C}_m}} y_{\mathbf{x}_m})(u) \triangleq \int_0^1 \mathbf{W}_{\mathbf{C}_m}(u,v) y_{\mathbf{x}_m}(v) \mathrm{d}v. \tag{38}$$

Thus, based on the discussion above, $T^k_{\mathbf{W}_{\mathbf{C}_m}} y_{\mathbf{x}_m}$ and $\mathbf{C}_m^k \mathbf{x}_m$ are operationally interchangeable. We can also define k-hop aggregation over $\mathbf{W}$ using the operator $T_{\mathbf{W}} y$ when y is related to $y_{\mathbf{x}_m}$ by $y_{\mathbf{x}_m}(\rho_i) = y(\rho_i)$, where ρ_i is defined in (18). Thus, graphon $\mathbf{W}$ and the continuous representation y can be seen as generative models for covariance matrix $\mathbf{C}_m$ and data point $\mathbf{x}_m$. This observation is in parallel to that in the context of graphs and graphons [50]. We denote the graphon filter for a set of filter taps $\mathcal{H} = \{h_k\}_{k=0}^K$ by $\Psi(y; \mathbf{W}, \mathcal{H}) : [0,1] \to \mathbb{R}$, which is defined as

$$\Psi(y; \mathbf{W}, \mathcal{H})(u) \triangleq \sum_{k=0}^{K} h_k (T^k_{\mathbf{W}} y)(u). \tag{39}$$

Similar to coVariance filter, the frequency response of a graphon filter can be characterized via using eigendecomposition of $\mathbf{W}$ in (39). Because $\mathbf{W}$ is bounded and symmetric, the spectral decomposition of $\mathbf{W}$ can be expressed as

$$\mathbf{W}(u,v) = \sum_{i \in \mathbb{Z} \setminus \{0\}} \eta_i \Gamma_i(u) \Gamma_i(v), \tag{40}$$

where $\eta_i, \forall i \in \mathbb{Z}\backslash\{0\}$ are eigenvalues and Γ_i are the eigensignals of $\mathbf{W}$. Therefore, (39) can be re-stated as

$$\Psi(y; \mathbf{W}, \mathcal{H})(u) = \sum_{i \in \mathbb{Z}\backslash\{0\}} \sum_{k=0}^{K} h_k \eta_i^k \Gamma_i(u) \int_0^1 \Gamma_i(v) y(v) \mathrm{d}v, \tag{41}$$

$$= \sum_{i \in \mathbb{Z}\backslash\{0\}} \tilde{h}(\eta_i) \Gamma_i(u) \int_0^1 \Gamma_i(v) y(v) \mathrm{d}v, \tag{42}$$

for $u \in [0,1]$. Note that (41) follows from (39) using (40) and (37), and we have used the definition $\tilde{h}(\eta) \triangleq \sum_{k=0}^{K} h_k \eta^k$ in (42). The term $\tilde{h}(\eta_i)$ characterizes the frequency response of a graphon filter and depends on the filter taps $\{h_k\}$ and the graphon eigenvalues.

B Proof of Theorem 2

In Theorem 2, we compare the continuous representations of the f-th outputs of VNNs $\Phi(\mathbf{x}_{m_1}; \mathbf{C}_{m_1}, \mathcal{H})$ and $\Phi(\mathbf{x}_{m_2}; \mathbf{C}_{m_2}, \mathcal{H})$. Our discussion in Appendix A showed that these continuous representations appear naturally as the outputs of the architectures $\tilde{\Phi}(y_{\mathbf{x}_{m_1}}; \mathbf{W}_{\mathbf{C}_{m_1}}, \mathcal{H})$ and $\tilde{\Phi}(y_{\mathbf{x}_{m_1}}; \mathbf{W}_{\mathbf{C}_{m_1}}, \mathcal{H})$ instantiated on graphon approximations $\mathbf{W}_{\mathbf{C}_{m_1}}$ and $\mathbf{W}_{\mathbf{C}_{m_2}}$, respectively. Therefore, our subsequent analysis is focused on the comparisons between their constituent graphon filters that eventually enables us to establish the convergence between f-th outputs of $\tilde{\Phi}(y_{\mathbf{x}_{m_1}}; \mathbf{W}_{\mathbf{C}_{m_1}}, \mathcal{H})$ and $\tilde{\Phi}(y_{\mathbf{x}_{m_1}}; \mathbf{W}_{\mathbf{C}_{m_1}}, \mathcal{H})$.

We begin by establishing various results pertaining to the comparisons between $\mathbf{W}$ and $\mathbf{W}_{\mathbf{C}_m}$, y and $y_{\mathbf{x}_m}$, and difference between eigenvalues of two distinct graphons. We leverage the (Ω, ζ)-dominant property of sequence of covariance matrices $\{\mathbf{C}_m\}$ in (20) and the Lipschitz condition of graphon in (21) to establish the following result.

Lemma 3: Given an α_1-Lipschitz graphon $\mathbf{W}$ and $\mathbf{W}_{\mathbf{C}_m}$ as graphon representation of a (Ω, ζ)-dominant covariance matrix $\mathbf{C}_m$, we have

$$\|\mathbf{W} - \mathbf{W}_{\mathbf{C}_m}\|_2 \leq \frac{\alpha_1 \Omega^{3/2}}{m^{3\zeta/2-1}}. \tag{43}$$

Proof: From the construction of $\mathbf{W}_{\mathbf{C}_m}$, we have

$$\|\mathbf{W} - \mathbf{W}_{\mathbf{C}_m}\|_2$$

$$= \left(\int_0^1 \int_0^1 \|\mathbf{W}(u,v) - \mathbf{W}_{\mathbf{C}_m}(u,v)\|^2 \mathrm{d}u\mathrm{d}v \right)^{\frac{1}{2}}, \tag{44}$$

$$= \left(\sum_{i,j} \int_{\mathcal{U}_i} \int_{\mathcal{U}_j} \|\mathbf{W}(u,v) - \mathbf{W}_{\mathbf{C}_m}(u,v)\|^2 \mathrm{d}u\mathrm{d}v \right)^{\frac{1}{2}}. \tag{45}$$

Without loss of generality, we assume that $\mathcal{U}_1 = [0, \rho_1]$ is the largest interval. Using the α_1-Lipschitz continuity of graphon $\mathbf{W}_{\mathbf{C}_m}$ and noting that $\mathbf{W}_{\mathbf{C}_m}(\rho_i, \rho_j) = \mathbf{W}(\rho_i, \rho_j)$, we have

$$\|\mathbf{W} - \mathbf{W}_{\mathbf{C}_m}\|_2 \leq \left(\sum_{i,j} \int_{I_i} \int_{I_j} \alpha_1^2 (|u| + |v|)^2 \mathrm{d}u\mathrm{d}v \right)^{\frac{1}{2}}, \tag{46}$$

$$\leq \left(m^2 \int_0^{\rho_1} \int_0^{\rho_1} \alpha_1^2 (|u| + |v|)^2 \mathrm{d}u\mathrm{d}v \right)^{\frac{1}{2}}, \tag{47}$$

$$\leq \left(m^2 \int_0^{\rho_1} \int_0^{\rho_1} \alpha_1^2 (|u| + |v|) \mathrm{d}u\mathrm{d}v \right)^{\frac{1}{2}}, \tag{48}$$

$$\leq \alpha m \rho_1^{3/2}. \tag{49}$$

Using the assumption that $\mathbf{C}_m$ is (Ω, ζ)-dominant, we have

$$\|\mathbf{W} - \mathbf{W}_{\mathbf{C}_m}\|_2 \leq \frac{\alpha_1 \Omega^{3/2}}{m^{3\zeta/2-1}}. \tag{50}$$

■

Next, we characterize the difference between a graphon signal $y \in L_2([0,1])$ and approximation $y_{\mathbf{x}_m}$ obtained from a random sample $\mathbf{x}$ in Lemma 4. For this purpose, we have the following assumption: a graphon signal y satisfies $|y(a) - y(b)| \leq \alpha_2 |a - b|, \forall a, b \in [0,1]$. We term a graphon signal satisfying this property as α_2-Lipschitz graphon signal.

Lemma 4: Given an α_2-Lipschitz graphon signal y and a graphon signal approximation $y_{\mathbf{x}_m}$ obtained from $\mathbf{x}_m \in \mathbb{R}^{m \times 1}$, we have

$$\|y - y_{\mathbf{x}_m}\|_2 \leq \frac{\alpha_2 \Omega^{3/2}}{m^{3\zeta/2-1}}. \tag{51}$$

Proof: Note that

$$\|y - y_{\mathbf{x}_m}\|_2 = \sum_{\mathcal{U}_i} \|y - y_{\mathbf{x}_m}\|_{L_2[I_i]}, \tag{52}$$

$$= \sum_{i=1}^{m} \left(\int_{\rho_{i-1}}^{\rho_i} (y(u) - y_{\mathbf{x}_m}(u))^2 \mathrm{d}u \right)^{\frac{1}{2}}, \tag{53}$$

where we have $\rho_0 = 0$. Using the Lipschitz property of graphon signal and (Ω, ζ)-property of $\mathbf{C}_m$, we have

$$\|y - y_{\mathbf{x}_m}\|_2 \leq m \left(\alpha_2^2 \int_0^{\rho_1} u^2 \mathrm{d}u \right)^{\frac{1}{2}} \leq \frac{\alpha_2 \Omega^{3/2}}{m^{3\zeta/2-1}}. \tag{54}$$

■

Next, we state Proposition 4 from [14] that characterizes a bound on the difference between eigenvalues from two graphons.

Lemma 5 (Proposition 4 from [14]): Consider two graphons $\mathbf{W}$ and $\mathbf{W}'$ with set of eigenvalues $\{\eta_i\}_{i=1}^{\infty}$ and $\{\beta_i\}_{i=1}^{\infty}$, respectively. Then, for all $i \in \mathbb{Z}^+$, we have $|\eta_i - \beta_i| \leq \|T_{\mathbf{W}-\mathbf{W}'}\|_2 \leq \|\mathbf{W} - \mathbf{W}'\|_2$.

We leverage Lemmas 3, 4, and 5 to bound the difference between graphon convolution $\Psi(y; \mathbf{W}, \mathcal{H})$ and convolution by the approximation $\Psi(y_{\mathbf{x}_m}; \mathbf{W}_{\mathbf{C}_m}, \mathcal{H})$ realized from the coVariance filter over $\mathbf{C}_m$.

In the following Lemma, we use the notations $\{\hat{\eta}_i\}$ and $\{\hat{\Gamma}_i\}$ for the set of eigenvalues and eigenfunctions, respectively, of $\mathbf{W}_{\mathbf{C}_m}$. The frequency response is assumed to be band-limited, such that, $|\tilde{h}(\eta)| = 0$ for $\eta \leq \eta_{\mathsf{c}}$. Furthermore, we assume that m_{c} largest eigenvalues of graphon $\mathbf{W}$ in terms of magnitude satisfy $|\eta| > \eta_{\mathsf{c}}$ and the set of such eigenvalues is denoted by $\mathcal{C}$.

Lemma 6 (Transferability of Graphon Filters): For a convolution $\Psi(y;\mathbf{W},\mathcal{H})$ and its approximation $\Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})$, under the assumptions A1-A5 and when Lemmas 3–5 hold, we have

$$\begin{aligned}&\|\Psi(y;\mathbf{W},\mathcal{H}) - \Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2\\ &\quad\leq \frac{\Omega^{3/2}}{m^{3\zeta/2-1}}\left(\alpha_2 + \alpha_1\left[\alpha_3 + \frac{\pi m_{\mathsf{c}}}{2\Delta_c}\right]\right),\end{aligned} \tag{55}$$

where $\Delta_c = \min_{i\neq j; i,j\in\mathcal{C}}\{|\eta_i - \hat{\eta}_j|\}$ and $\|y\|_2 \leq 1$.

Proof: Note that we can rewrite $\|\Psi(y;\mathbf{W},\mathcal{H}) - \Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2$ as

$$\begin{aligned}&\|\Psi(y;\mathbf{W},\mathcal{H}) - \Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2\\ &\quad= \|\Psi(y;\mathbf{W},\mathcal{H}) - \Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\\ &\qquad+ \Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H}) - \Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2,\end{aligned} \tag{56}$$

and using triangle inequality, we have

$$\begin{aligned}&\|\Psi(y;\mathbf{W},\mathcal{H}) - \Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2\\ &\quad\leq \underbrace{\|\Psi(y;\mathbf{W},\mathcal{H}) - \Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2}_{\text{Term 1}}\\ &\quad+ \underbrace{\|\Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H}) - \Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2}_{\text{Term 2}}.\end{aligned} \tag{57}$$

Next, we analyze Terms 1 and 2 from (57) separately.

Analysis of Term 1: Using the expansion of $\Psi(y;\mathbf{W},\mathcal{H})$ and $\Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})$, we have

$$\|\Psi(y;\mathbf{W},\mathcal{H}) - \Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2 = \left(\int_0^1 f^2(v)\mathsf{d}v\right)^{1/2}, \tag{58}$$

where

$$\begin{aligned}&f(v)\\ &\quad= \sum_{i=\in\mathbb{Z}\setminus\{0\}}\left[\tilde{h}(\eta_i)\Gamma_i(v)\int_0^1 y(u)\Gamma_i(u)\mathsf{d}u - \tilde{h}(\hat{\eta}_i)\hat{\Gamma}_i(v)\right.\\ &\qquad\left.\times\int_0^1 y(u)\hat{\Gamma}_i(u)\mathsf{d}u\right].\end{aligned} \tag{59}$$

By adding and subtracting $\tilde{h}(\hat{\eta}_i)\Gamma_i(v)\int_0^1 y(u)\Gamma_i(u)du$ in (59) and using the triangle inequality, we obtain

$$\begin{aligned}&\|\Psi(y;\mathbf{W},\mathcal{H}) - \Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2\\ &\quad\leq \left(\int_0^1 f_1^2(v)\mathsf{d}v\right)^{1/2} + \left(\int_0^1 f_2^2(v)\mathsf{d}v\right)^{1/2}\\ &\quad= \|f_1\|_2 + \|f_2\|_2,\end{aligned} \tag{60}$$

where

$$f_1(v) = \sum_{i\in\mathbb{Z}\setminus\{0\}}\left[(\tilde{h}(\eta_i) - \tilde{h}(\hat{\eta}_i))\Gamma_i(v)\int_0^1 y(u)\Gamma_i(u)\mathsf{d}u\right], \tag{61}$$

and

$$f_2(v) = \sum_{i\in\mathbb{Z}\setminus\{0\}}\left[\tilde{h}(\hat{\eta}_i)\Gamma_i(v)\int_0^1 y(u)(\Gamma_i(u) - \hat{\Gamma}_i(u))\mathsf{d}u\right]. \tag{62}$$

Using the Lipschitz property of graphon filter, we have $|\tilde{h}(\eta_i) - \tilde{h}(\hat{\eta}_i)| \leq \alpha_3|\eta_i - \hat{\eta}_i|$. Therefore, Lemma 5 and Lemma 3 lead to

$$\|f_1\|_2 \leq \frac{\alpha_3\alpha_1\Omega^{3/2}}{m^{3\zeta/2-1}}. \tag{63}$$

for any y that satisfies $\|y\|_2 \leq 1$. To analyze $\|f_2\|_2$, we leverage the Cauchy-Schwarz inequality to have

$$\begin{aligned}\|f_2\|_2 &\leq \sum_{i\in\mathbb{Z}\setminus\{0\}}|\tilde{h}(\hat{\eta}_i)|\|\Gamma_i\|_2\|y(\Gamma_i - \hat{\Gamma}_i)\|_2, &(64)\\ &\leq \sum_{i\in\mathbb{Z}\setminus\{0\}}|\tilde{h}(\hat{\eta}_i)|\|\Gamma_i - \hat{\Gamma}_i\|_2, &(65)\end{aligned}$$

where (65) follows from (64), without loss of generality for $\|y\|_2 = 1$, $\|\Gamma_i\|_2 = 1$ and another application of Cauchy-Schwarz inequality. Next, we note that the integral operator $T_{\mathbf{W}}$, such that, $(T_{\mathbf{W}}y)(v) = \int_0^1 \mathbf{W}(u,v)y(u)\mathsf{d}u$ is a self-adjoint Hilbert-Schmidt operator and $\mathbf{W}$ admits the spectral decomposition with $\{\eta_i\}$ as eigenvalues and $\{\Gamma_i\}$ as eigensignals. Therefore, to analyze $\|\Gamma_i - \hat{\Gamma}_i\|_2$, we note that Γ_i is projection of operator $T_{\mathbf{W}}$ associated with eigenvalue η_i and $\hat{\Gamma}_i$ is projection of operator $T_{\mathbf{W}_{\mathbf{C}_m}}$ associated with eigenvalue $\hat{\eta}_i$. By dividing the spectrum of $T_{\mathbf{W}}$ as $\mathsf{spec}(T_{\mathbf{W}}) = \{\eta_i\} \cup \{\eta_j\}_{j\neq i}$ and that of $T_{\mathbf{W}_{\mathbf{C}_m}}$ as $\mathsf{spec}(T_{\mathbf{W}_{\mathbf{C}_m}}) = \{\hat{\eta}_i\} \cup \{\hat{\eta}_j\}_{j\neq i}$, we apply Proposition 2.3 from [67] to have

$$\|\Gamma_i - \hat{\Gamma}_i\|_2 \leq \frac{\pi}{2}\frac{\|T_{\mathbf{W}} - T_{\mathbf{W}_{\mathbf{C}_m}}\|_2}{d_i}, \tag{66}$$

where $d_i > 0$ is a constant that satisfies $|\eta_i - \hat{\eta}_{i+1}| \geq d_i$, $|\eta_i - \hat{\eta}_{i-1}| \geq d_i$, $|\eta_{i+1} - \hat{\eta}_i| \geq d_i$, and $|\eta_{i-1} - \hat{\eta}_i| \geq d_i$. Using (66), Lemma 5 and Lemma 3 in (65), we have

$$\|f_2\|_2 \leq \frac{\pi\alpha\Omega^{3/2}}{2\Delta_c m^{3\zeta/2-1}}\sum_{i\in\mathbb{Z}\setminus 0}|\tilde{h}(\hat{\eta}_i)|, \tag{67}$$

where $\Delta_c = \min_i d_i$. Next, we note that $|\tilde{h}(\hat{\eta}_i)| \leq 1$ under Assumption **A5** as all eigenvalues of $\mathbf{W}_{\mathbf{C}_m}$ are smaller than or equal to 1. Hence, we have $\sum_{i\in\mathbb{Z}\setminus 0}|\tilde{h}(\hat{\eta}_i)| \leq m_{\mathsf{c}}$ under the band-limiting condition $\tilde{h}(\hat{\eta}_i) = 0$ for $|\eta_i| \leq \eta_c$ and $\tilde{h}(\hat{\eta}_i) \neq 0$ for at most m_c eigenvalues. In this scenario, we can rewrite (67) as

$$\|f_2\|_2 \leq \frac{\pi\alpha_1\Omega^{3/2}m_{\mathsf{c}}}{2\Delta_c m^{3\zeta/2-1}}. \tag{68}$$

Clearly, there is a trade-off between m_c and ζ as we must have $m_c < m^{3\zeta/2-1}$ and $\zeta > 2/3$ for (68) to have decreasing behavior in m. Equations (63) and (68) provide the upper bound on Term 1.

Analysis of Term 2:. We can expand term 2 as

$$\|\Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})-\Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2=\left(\int_0^1 g^2(v)\mathrm{d}v\right)^{1/2}, \tag{69}$$

where

$$g(v)=\sum_{i=1}^{\infty}\left[\tilde{h}(\eta_i)\hat{\Gamma}_i(v)\int_0^1(y(u)-y_{\mathbf{x}_m}(u))\hat{\Gamma}_i(u)\mathrm{d}u\right]. \tag{70}$$

Therefore, using (70), we have

$$\begin{aligned}&\|\Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})-\Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2\\&\quad=\|\Psi(y-y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2.\end{aligned} \tag{71}$$

Note that for a frequency response that satisfies $\tilde{h}(\eta)\le 1$, the graphon filter is non-expanding and therefore, we have

$$\|\Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})-\Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2\le\|y-y_{\mathbf{x}_m}\|_2. \tag{72}$$

Using Lemma 4, we have

$$\|\Psi(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})-\Psi(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})\|_2\le\frac{\alpha_2\Omega^{3/2}}{m^{3\zeta/2-1}}. \tag{73}$$

Therefore, by combining the upper bounds on Term 1 and Term 2 from (63), (68), and (73), the proof of Lemma 6 is concluded.■

Lemma 6 establishes the transference between the graphon $\mathbf{W}$ and the graphon approximation $\mathbf{W}_{\mathbf{C}_m}$ obtained from the covariance matrix $\mathbf{C}_m$. We leverage the result in Lemma 6 to establishing the transference for graphon neural networks in a similar setting. We denote the f-th output for graphon neural network $\tilde{\Psi}(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})$ with F outputs in the final layer by $[\tilde{\Psi}(y;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})]_f$.

Lemma 7 (Transferability of Graphon Neural Networks): Consider a graphon neural network $\tilde{\Phi}(\cdot;\mathbf{W},\mathcal{H})$ with L layers and F outputs per layer and a VNN $\Phi(\cdot;\mathbf{C}_m,\mathcal{H})$ with graphon neural network representation as $\tilde{\Phi}(\cdot;\mathbf{W}_{\mathbf{C}_m},\mathcal{H})$. If the covariance matrix $\mathbf{C}_m$ belongs to a (Ω,ζ)-dominant sequence of covariance matrices and its graphon approximation $\mathbf{W}_{\mathbf{C}_m}$ belongs to a graphon family of α-Lipschitz graphon $\mathbf{W}$, then under the assumptions A1-A5, for $\|y\|_2\le 1$ and $2/3<\zeta\le 1$, we have

$$\begin{aligned}&\|[\tilde{\Phi}(y;\mathbf{W},\mathcal{H})]_f-[\tilde{\Phi}(y_{\mathbf{x}_m};\mathbf{W}_{\mathbf{C}_m},\mathcal{H})]_f\|_2\\&\quad\le LF^L\left(\frac{\Omega^{3/2}}{m^{3\zeta/2-1}}\left[\alpha_2+\alpha_1\left[\alpha_3+\frac{\pi m_c}{2\Delta_c}\right]\right]\right).\end{aligned} \tag{74}$$

The proof of Lemma 7 leverages Lemma 6 and accommodates the impact of multi-layer VNN architecture. We refer the reader to (23)–(28) in [14] for exact analytical steps. Finally, by applying the triangle inequality on (74), we establish the transference between graphon neural network approximations $\tilde{\Phi}(\cdot;\mathbf{W}_{\mathbf{C}_{m_1}},\mathcal{H})$ and $\tilde{\Phi}(\cdot;\mathbf{W}_{\mathbf{C}_{m_2}},\mathcal{H})$ for VNNs $\Phi(\cdot;\mathbf{C}_{m_1},\mathcal{H})$ and $\Phi(\cdot;\mathbf{C}_{m_2},\mathcal{H})$, respectively, and the proof of Theorem 2 is concluded.

References

[1] P. Liu, “A survey of remote-sensing Big Data,” *Front. Environ. Sci.*, vol. 3, 2015, Art. no. 45.

[2] B. H. Brinkmann, M. R. Bower, K. A. Stengel, G. A. Worrell, and M. Stead, “Large-scale electrophysiology: Acquisition, compression, encryption, and storage of Big Data,” *J. Neurosci. Methods*, vol. 180, no. 1, pp. 185–192, 2009.

[3] J. Cai, J. Luo, S. Wang, and S. Yang, “Feature selection in machine learning: A new perspective,” *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[4] A. Y. Ng, “Feature selection, L1 vs. L2 regularization, and rotational invariance,” in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 78–86.

[5] S. Sihag, G. Mateos, C. McMillan, and A. Ribeiro, “coVariance neural networks,” in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2022, pp. 17003–17016.

[6] A. C. Evans, “Networks of anatomical covariance,” *Neuroimage*, vol. 80, pp. 489–504, 2013.

[7] H. Murase and S. K. Nayar, “Illumination planning for object recognition using parametric eigenspaces,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 12, pp. 1219–1227, Dec. 1994.

[8] D. Stephenson, “Correlation of spatial climate/weather maps and the advantages of using the Mahalanobis metric in predictions,” *Tellus A*, vol. 49, no. 5, pp. 513–527, 1997.

[9] H. Shao, W. H. Lam, A. Sumalee, A. Chen, and M. L. Hazelton, “Estimation of mean and covariance of peak hour origin–destination demands from day-to-day traffic counts,” *Transp. Res. B-Meth.*, vol. 68, pp. 52–75, 2014.

[10] M. N. Ismail, A. Aborujilah, S. Musa, and A. Shahzad, “Detecting flooding based DoS attack in cloud computing environment using covariance matrix approach,” in *Proc. Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2013, pp. 1–6.

[11] S. Verma and Z.-L. Zhang, “Stability and generalization of graph convolutional neural networks,” in *Proc. ACM Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1539–1548.

[12] N. Keriven, A. Bietti, and S. Vaiter, “Convergence and stability of graph convolutional networks on large random graphs,” in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 21512–21523.

[13] F. Gama, J. Bruna, and A. Ribeiro, “Stability properties of graph neural networks,” *IEEE Trans. Signal Process.*, vol. 68, pp. 5680–5695, 2020.

[14] L. Ruiz, L. Chamon, and A. Ribeiro, “Graphon neural networks and the transferability of graph neural networks,” in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1702–1712.

[15] I. T. Jolliffe and B. Morgan, “Principal component analysis and exploratory factor analysis,” *Statist. Methods Med. Res.*, vol. 1, no. 1, pp. 69–95, 1992.

[16] E. Elhaik, “Principal component analyses (PCA)-based findings in population genetic studies are highly biased and must be reevaluated,” *Sci. Rep.*, vol. 12, no. 1, pp. 1–35, 2022.

[17] R. Levie, W. Huang, L. Bucci, M. M. Bronstein, and G. Kutyniok, “Transferability of spectral graph convolutional neural networks,” *J. Mach. Learn. Res.*, vol. 22, no. 272, pp. 1–59, 2021.

[18] Q. Zhu, C. Yang, Y. Xu, H. Wang, C. Zhang, and J. Han, “Transfer learning of graph neural networks with ego-graph inf. maximization,” in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 1766–1779.

[19] S. Maskey, R. Levie, and G. Kutyniok, “Transferability of graph neural networks: An extended graphon approach,” *Appl. Comput. Harmon. Anal.*, vol. 63, pp. 48–83, 2023.

[20] Y. Yang et al., “Data-efficient brain connectome analysis via multi-task meta-learning,” in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 4743–4751.

[21] R. D. Markello et al., “Neuromaps: Structural and functional interpretation of brain maps,” *Nature Methods*, vol. 19, no. 11, pp. 1472–1479, 2022.

[22] A. Schaefer et al., “Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity MRI,” *Cereb. Cortex*, vol. 28, no. 9, pp. 3095–3114, 2018.

[23] P. Hagmann et al., “Mapping the structural core of human cerebral cortex,” *PLoS Biol.*, vol. 6, no. 7, 2008, Art. no. e159.

[24] Y. Zeighami and A. C. Evans, “Association vs prediction: The impact of cortical surface smoothing and parcellation on brain age,” *Front. Big Data*, vol. 4, 2021, Art. no. 637724.

[25] A. I. Luppi and E. A. Stamatakis, “Combining network topology and information theory to construct representative brain networks,” *Netw. Neurosci.*, vol. 5, no. 1, pp. 96–124, 2021.

[26] J. Royer et al., “An open MRI dataset for multiscale neuroscience,” *Sci. Data*, vol. 9, no. 1, pp. 1–12, 2022.

[27] B. S. Khundrakpam et al., "Prediction of brain maturity based on cortical thickness at different spatial resolutions," *Neuroimage*, vol. 111, pp. 350–359, 2015.

[28] S. Sihag, G. Mateos, C. McMillan, and A. Ribeiro, "Explainable brain age prediction using covariance neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2023, pp. 46958–46988. [Online]. Available: https://openreview.net/forum?id=cAhJF87GN0

[29] J. H. Cole and K. Franke, "Predicting age using neuroimaging: Innovative brain ageing biomarkers," *Trends Neurosci.*, vol. 40, no. 12, pp. 681–690, 2017.

[30] J. H. Cole et al., "Brain age predicts mortality," *Mol. Psychiatry*, vol. 23, no. 5, pp. 1385–1392, 2018.

[31] L. Baecker, R. Garcia-Dias, S. Vieira, C. Scarpazza, and A. Mechelli, "Machine learning for brain age prediction: Introduction to methods and clinical applications," *EBioMedicine*, vol. 72, 2021, Art. no. 103600.

[32] L. Baecker et al., "Brain age prediction: A comparison between machine learning models using region-and voxel-based morphometric data," *Hum. Brain Mapping*, vol. 42, no. 8, pp. 2332–2346, 2021.

[33] K. Franke and C. Gaser, "Ten years of BrainAGE as a neuroimaging biomarker of brain aging: What insights have we gained?," *Front. Neurol.*, vol. 10, 2019, Art. no. 789.

[34] K. Franke and C. Gaser, "Longitudinal changes in individual brainage in healthy aging, mild cognitive impairment, and Alzheimer's disease," *GeroPsych: J. Gerontopsychol. Geriatr. Psychiatry*, vol. 25, no. 4, 2012, Art. no. 235.

[35] C. Yin et al., "Anatomically interpretable deep learning of brain age captures domain-specific cognitive impairment," *Proc. Nat. Acad. Sci.*, vol. 120, no. 2, 2023, Art. no. e2214634120.

[36] A. Lombardi et al., "Explainable deep learning for personalized age prediction with brain morphology," *Front. Neurosci.*, vol. 15, 2021, Art. no. 578.

[37] R. J. Jirsaraie et al., "A systematic review of multimodal brain age studies: Uncovering a divergence between model accuracy and utility," *Patterns*, vol. 4, no. 4, 2023, Art. no. 100712.

[38] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[39] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 9525–9536.

[40] E. Lee, D. Braines, M. Stiffler, A. Hudler, and D. Harborne, "Developing the sensitivity of lime for better machine learning explanation," in *Proc. SPIE*, vol. 11006, pp. 349–356, 2019.

[41] E. Black, M. Raghavan, and S. Barocas, "Model multiplicity: Opportunities, concerns, and solutions," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, 2022, pp. 850–863.

[42] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan, "L-shapley and C-shapley: Efficient mxodel interpretation for structured data," in *Proc. Int. Conf. Learn. Representations*, 2019.

[43] S. Sihag, G. Mateos, C. McMillan, and A. Ribeiro, "Predicting brain age using transferable coVariance neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2023, pp. 1–5.

[44] J. Shlens, "A tutorial on principal component analysis," 2014, *arXiv:1404.1100*.

[45] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[46] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: A comprehensive review," *Comput. Soc. Netw.*, vol. 6, no. 1, pp. 1–23, 2019.

[47] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal process on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[48] A. Sandryhaila and J. M. Moura, "Discrete signal process. on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[49] A. Loukas, "How close are the eigenvectors of the sample and actual covariance matrices?," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2228–2237.

[50] L. Ruiz, L. F. Chamon, and A. Ribeiro, "Graphon signal processing," *IEEE Trans. Signal Process.*, vol. 69, pp. 4961–4976, 2021.

[51] C. Borgs, J. T. Chayes, L. Lovász, V. T. Sós, and K. Vesztergombi, "Convergent sequences of dense graphs I: Subgraph frequencies, metric properties and testing," *Adv. Math.*, vol. 219, no. 6, pp. 1801–1851, 2008.

[52] L. Lovász, *Large Networks and Graph Limits*, Providence, RI, 1423 USA: Amer. Math. Soc., 2012.

[53] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 4768–4777.

[54] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should i trust you?" Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.

[55] N. J. Tustison et al., "Large-scale evaluation of ANTs and freesurfer cortical thickness measurements," *Neuroimage*, vol. 99, pp. 166–179, 2014.

[56] S. R. Das, B. B. Avants, M. Grossman, and J. C. Gee, "Registration based cortical thickness measurement," *Neuroimage*, vol. 45, no. 3, pp. 867–879, 2009.

[57] S. Sihag, G. Mateos, and A. Ribeiro, "Towards a foundation model for brain age prediction using covariance neural networks," 2024, *arXiv:2402.07684*.

[58] N. Alon and A. Naor, "Approximating the cut-norm via Grothendieck's inequality," *SIAM J. Comput.*, vol. 35, no. 4, pp. 787–803, 2006.

[59] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 8026–8037.

[60] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2623–2631.

[61] M. Jové, M. Portero-Otín, A. Naudí, I. Ferrer, and R. Pamplona, "Metabolomics of human brain aging and age-related neurodegenerative diseases," *J. Neuropathol. Exp. Neurol.*, vol. 73, no. 7, pp. 640–657, 2014.

[62] T. Schäfer and C. Ecker, "fsbrain: An R package for the visualization of structural neuroimaging data," *bioRxiv 2020.09. 18.302935*, 2020.

[63] I. Beheshti, S. Nugent, O. Potvin, and S. Duchesne, "Bias-adjustment in neuroimaging-based brain age frameworks: A robust scheme," *Neuroimage Clin.*, vol. 24, 2019, Art. no. 102063.

[64] A.-M. G. de Lange and J. H. Cole, "Commentary: Correction procedures in brain-age prediction," *Neuroimage Clin.*, vol. 26, 2020, Art. no. 102229.

[65] J. Bien and R. J. Tibshirani, "Sparse estimation of a covariance matrix," *Biometrika*, vol. 98, no. 4, pp. 807–820, 2011.

[66] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2020.

[67] A. Seelmann, "Notes on the $\sin 2\Theta$ theorem," *Integral Equ. Oper. Theory.*, vol. 79, no. 4, pp. 579–597, 2014.