Disentangling Neurodegeneration with Brain Age Gap Prediction Models: A Graph Signal Processing Perspective

Saurabh Sihag[1], Gonzalo Mateos[2], and Alejandro Ribeiro[3]

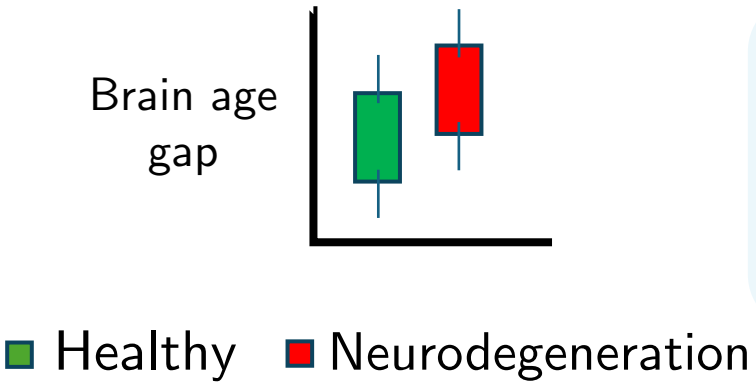
[1]University at Albany – ssihag@albany.edu
[2]University of Rochester – gmateosb@ece.rochester.edu
[3]University of Pennsylvania – aribeiro@seas.upenn.edu

Brain age gap

- Individual rate of “aging” is different from chronological rate of aging
  - Driven by environment, genetics, neurodegeneration
- Brain age provides a biological estimate brain age, derived from **neuroimaging**

Brain age gap

- Individual rate of “aging” is different from chronological rate of aging
  - Driven by environment, genetics, neurodegeneration
- Brain age provides a biological estimate brain age, derived from **neuroimaging**
- The brain age gap is the **deviation** between brain age and chronological age

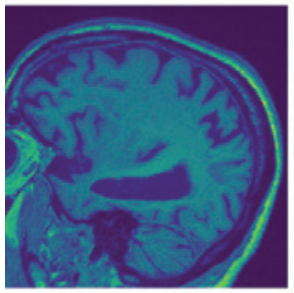


Brain age gap $\propto$ individual risks for neurological, neuropsychiatric and neurodegenerative diseases

Neurodegeneration (in terms of cortical atrophy)

- Neurodegeneration is **accelerated decline** of structure or function of the brain
- **Cortical atrophy**: reduction in cortical **thickness**/volume/area

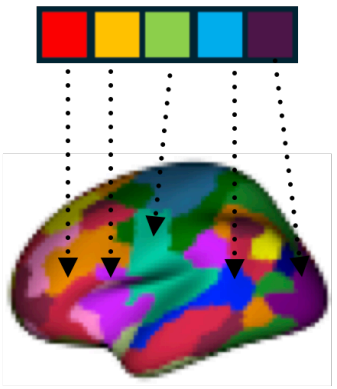
(characteristic of healthy aging and disorders like Alzheimer's disease (AD), frontotemporal dementia (FTD), etc.)



MRI scan

Preprocessing
(e.g., Freesurfer)

$\mathbf{x} = [x_1, \ldots, x_m]$

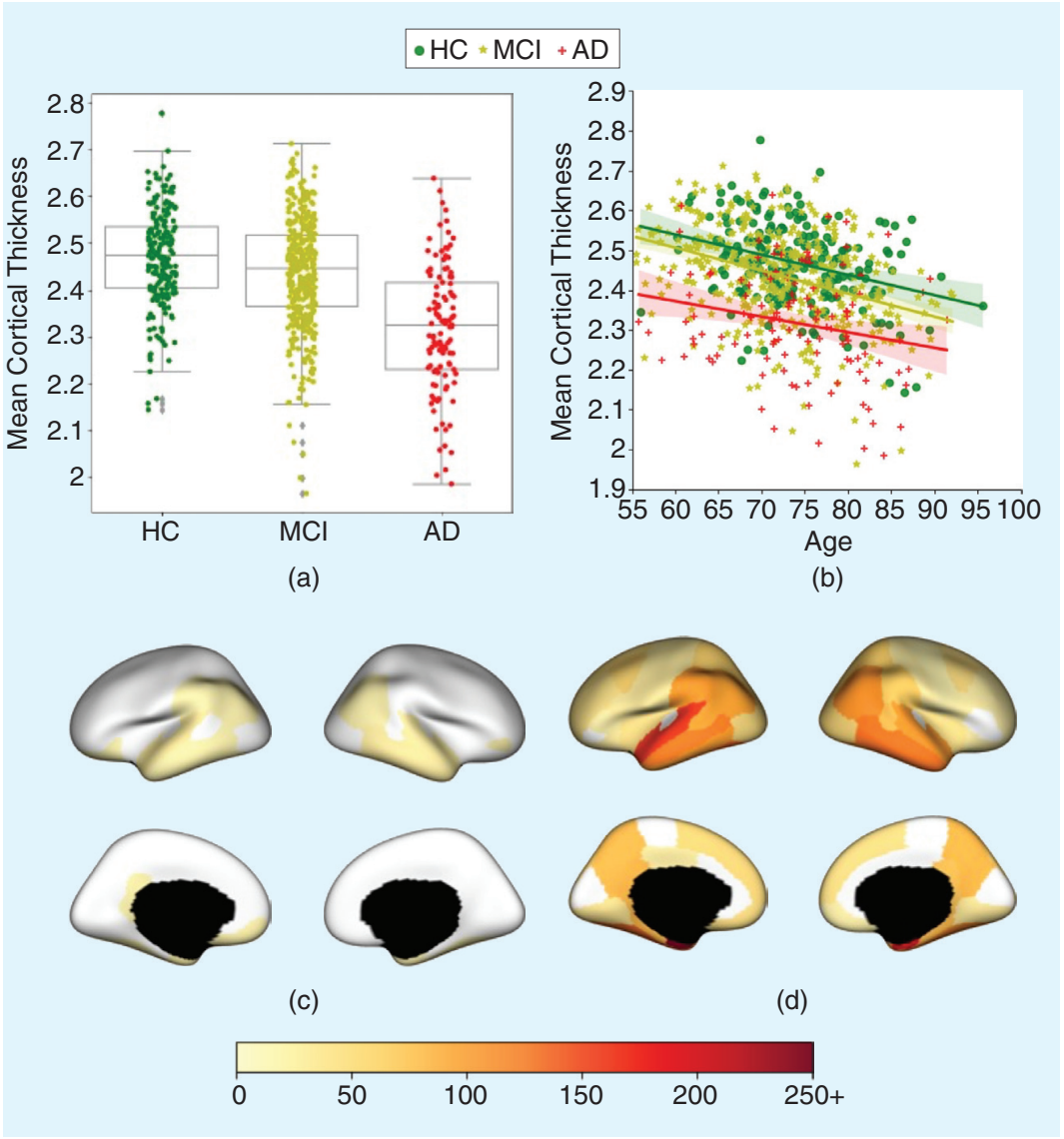


Cortical thickness features

x_i is cortical thickness for brain region i

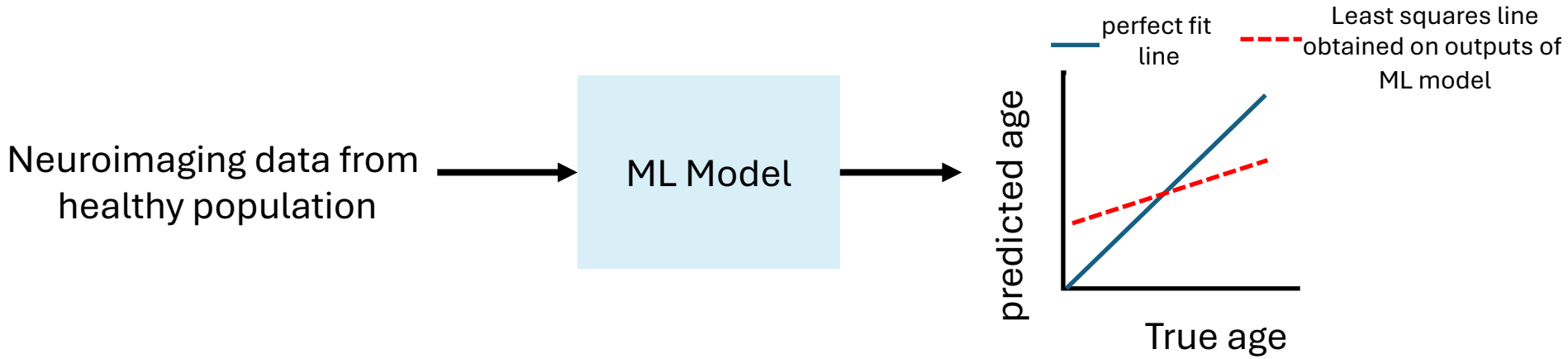
Case study (Neurodegeneration)

- **Data:** cortical thickness from 3 cohorts
  - HC (healthy)
  - MCI (Mild cognitive impairment)
  - AD (Alzheimer's disease)
- Larger **cortical atrophy** is feature of AD
- MCI is precursor to AD
  ⇨ shows intermediate cortical atrophy between HC and AD
- **Aging** also leads to cortical atrophy



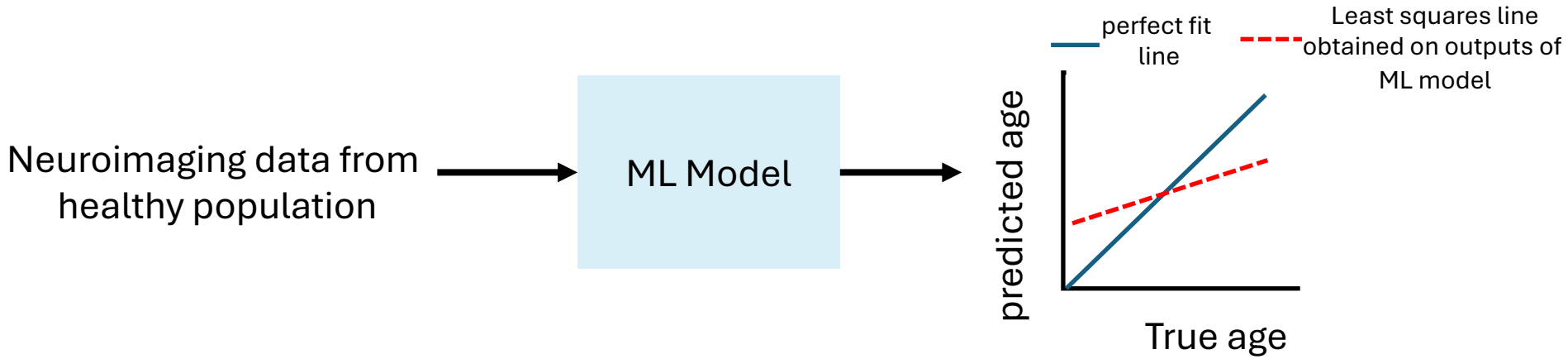
Brain age gap evaluation using ML

Step 1. Train ML model to predict chronological age for healthy controls from cortical thickness features



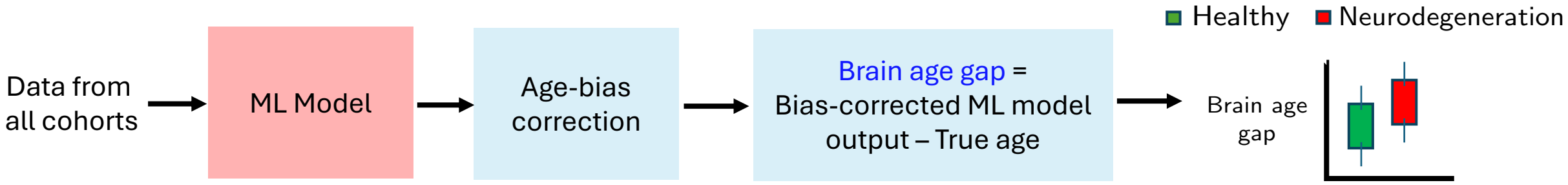
Brain age gap evaluation using ML

Step 1. Train ML model to predict chronological age for healthy controls from cortical thickness features



Step 2. Linear regression-based age-bias correct for outputs of ML model

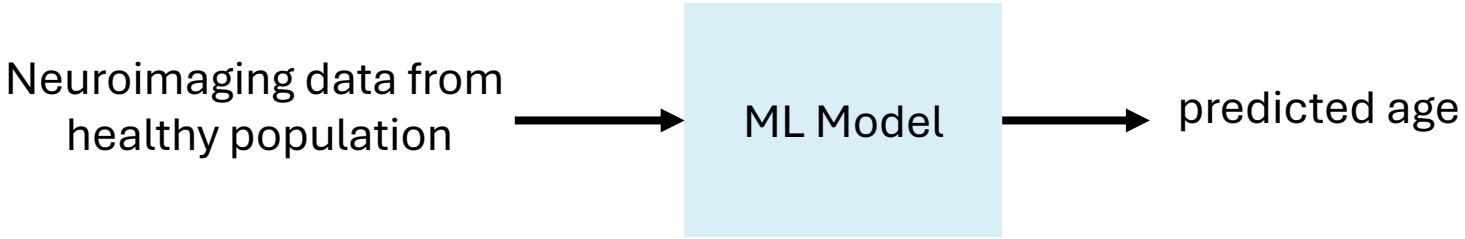
Step 3. Obtain brain age gap for healthy controls and individuals with neurodegenerative condition.



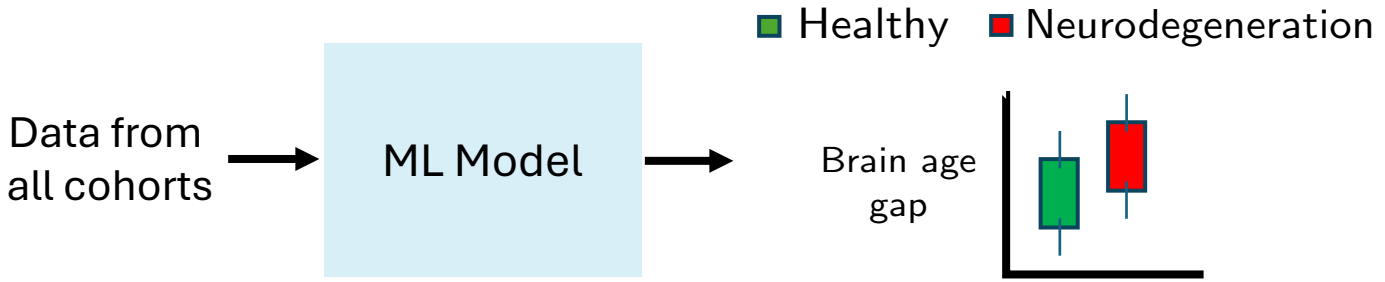
Brain age gap prediction is a transfer learning problem

- Train ML model to predict age on a **large** dataset (healthy population)

Pre-training



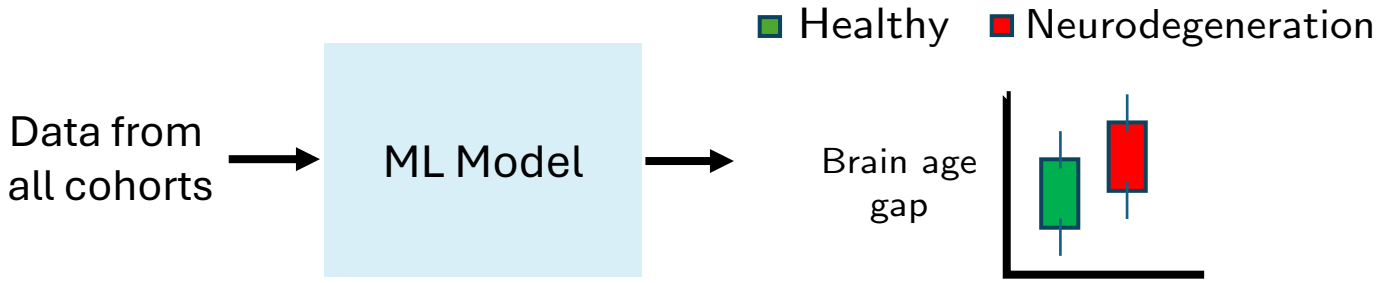
- Apply the **pre-trained** ML model on a target dataset (neurodegeneration)



- Brain age gap is the **residual** of the model

Brain age gap prediction is a transfer learning problem

- **Some observations about a meaningful brain age gap**
  - We expect model performance to **degrade** in target population
    - ✓ Degradation in performance (residuals) in a **specific direction**
    - ✓ Degradation in performance (residuals) $\propto$ **disease severity/status**

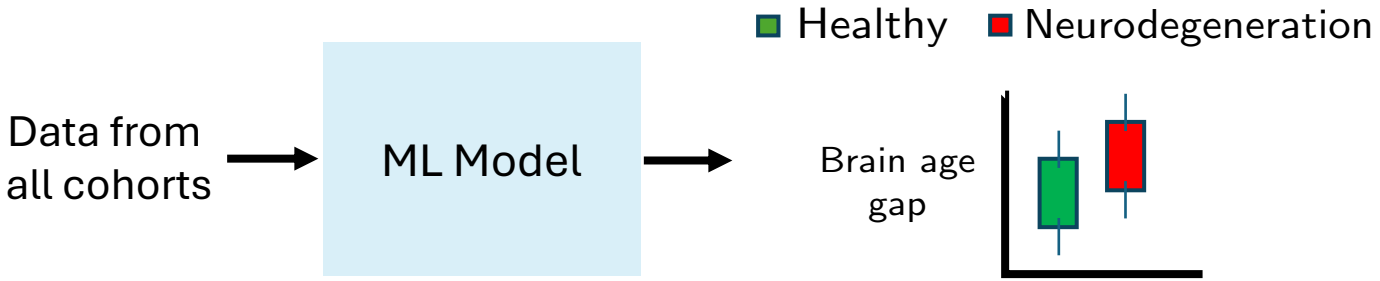


Choice of learning parametrization

- Choice of ML model dictates how data is leveraged to gauge brain age gap

- Prevalent approaches focus on achieving *perfect* pre-training performance
  - **Performance-driven approaches**

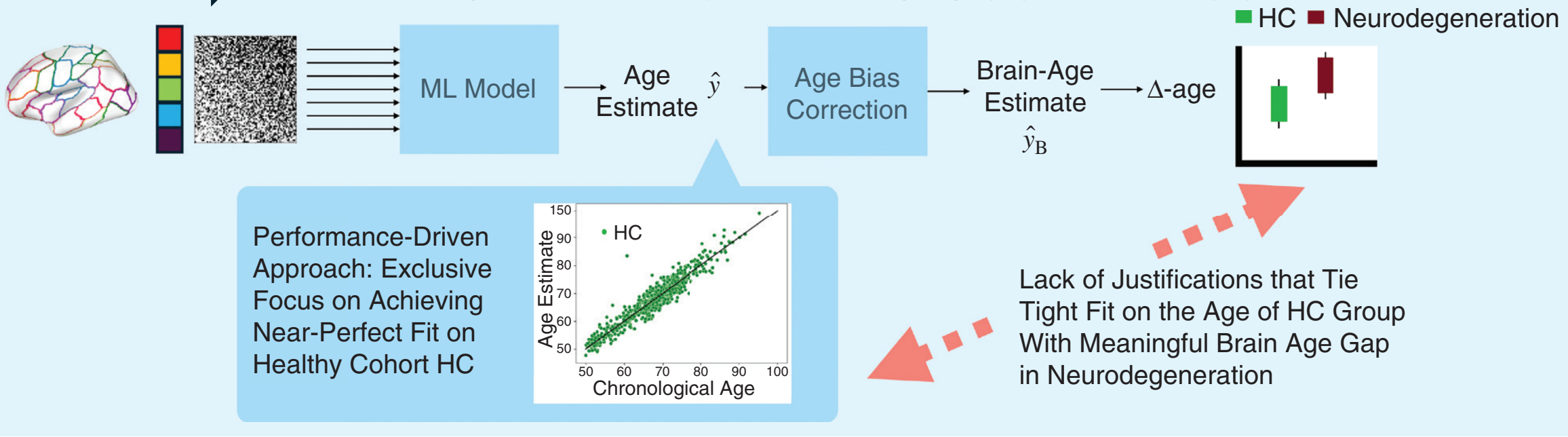
- **Performance-driven approaches** do not guarantee `**meaningful**' brain age gap



Choice of learning parametrization

- Neural networks are prevalent in performance-driven approaches
- A Neural Network may **not be interpretable** and prone to **overfitting**

methodological obscurity in brain age gap prediction pipeline



Choice of learning parametrization

- Neural networks are prevalent in performance-driven approaches
- A Neural Network may **not be interpretable** and prone to **overfitting**

⇨ **methodological obscurity** in brain age gap prediction pipeline

Performance in pre-training does not dictate **meaningful residuals** in target population

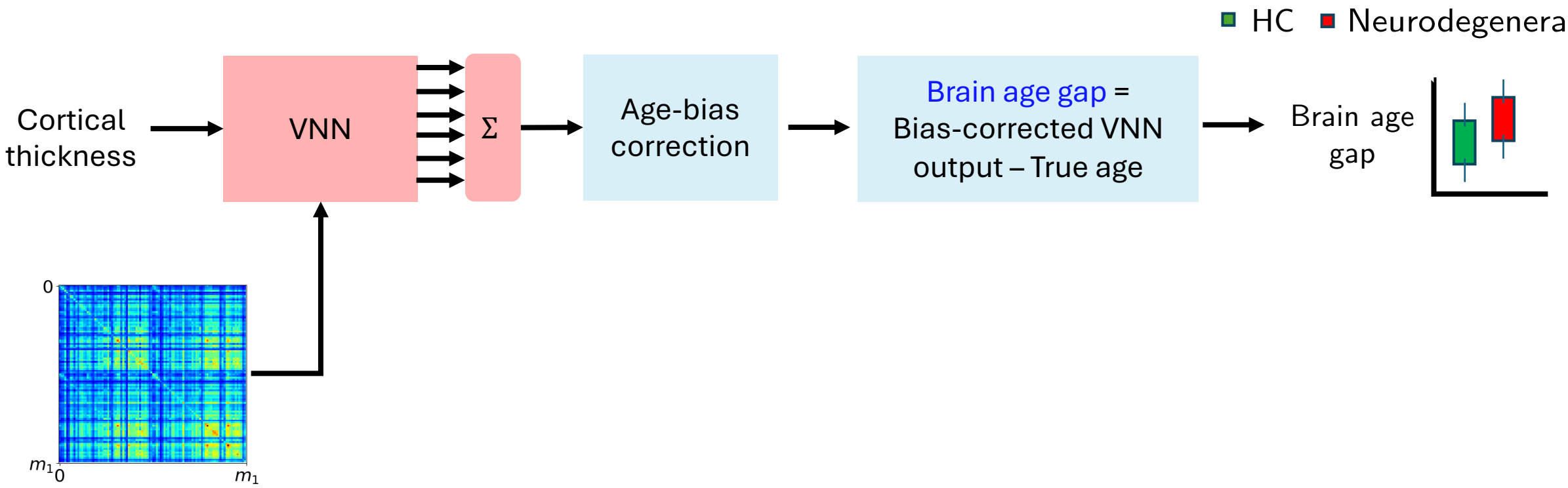
A principled approach to brain age gap prediction

- **Focus on residuals** of the ML model, not prediction performance
- **Qualitative evaluation** during pre-training

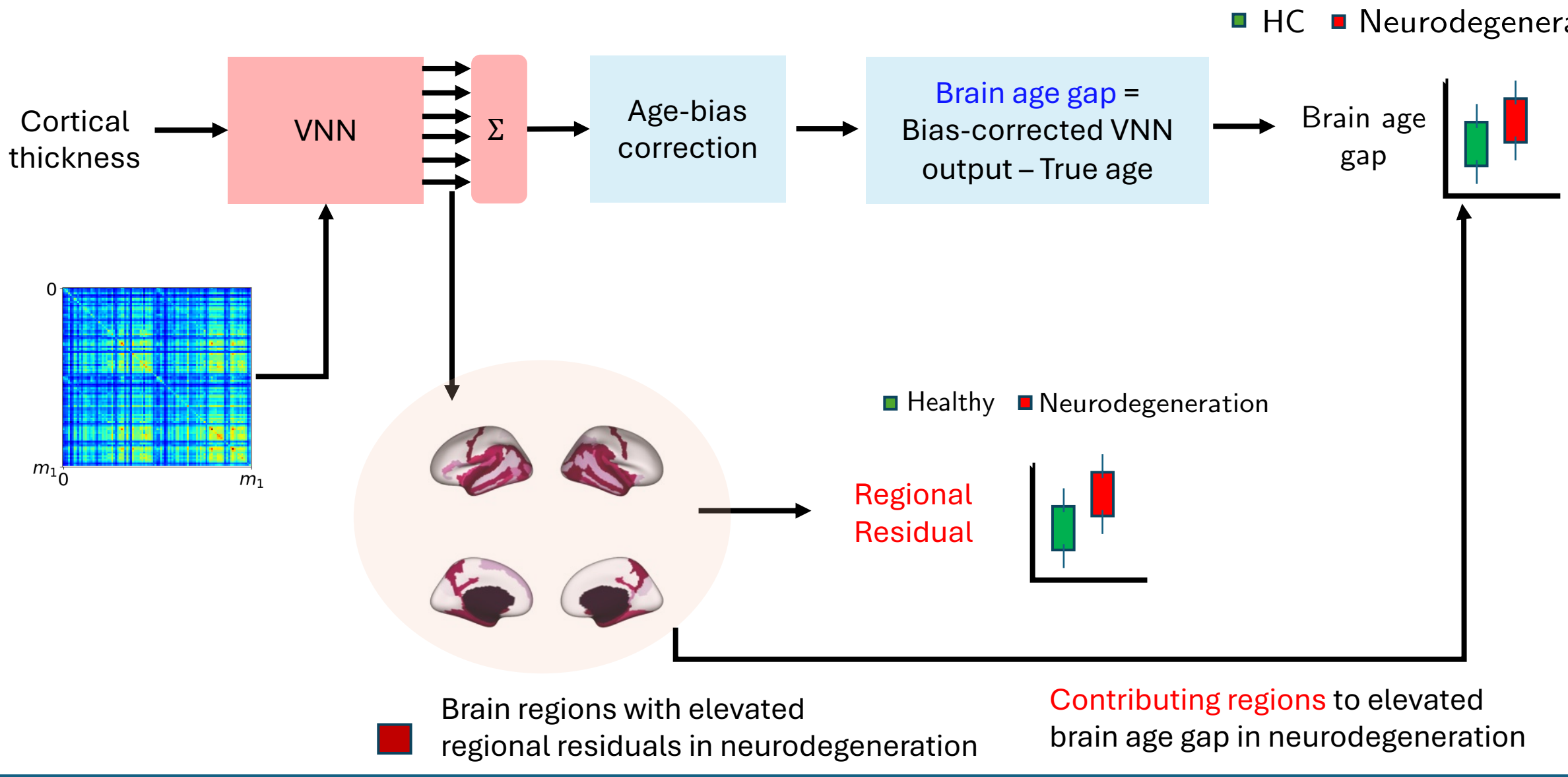
  what does the model learn during pre-training on healthy population?
- **Interpretability/explainability:**

  what's driving elevated brain age gap (residuals) in neurodegeneration?
- **Generalizability** to diverse target populations

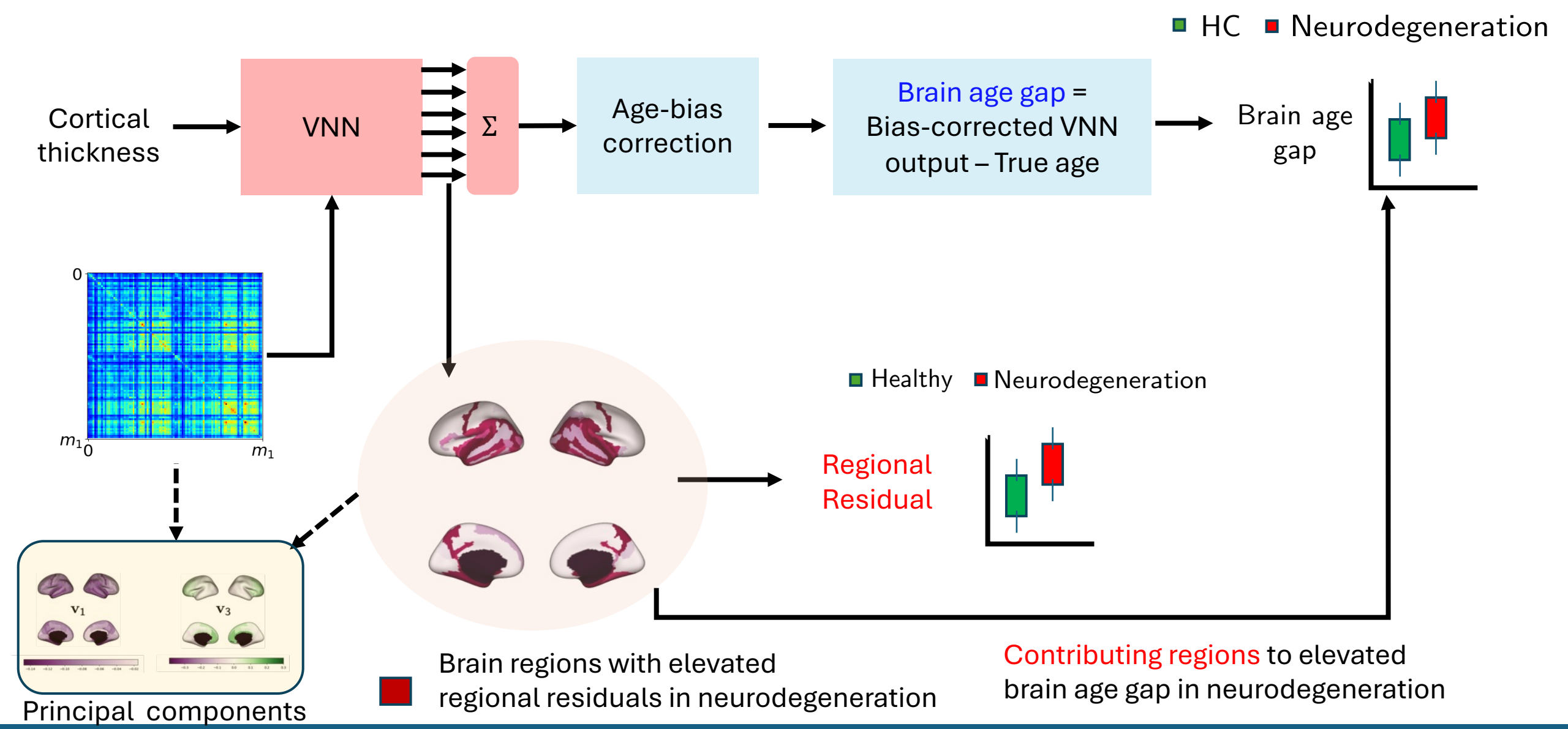
VNNs provide an anatomically interpretable and explainable brain age gap



VNNs provide an anatomically interpretable and explainable brain age gap

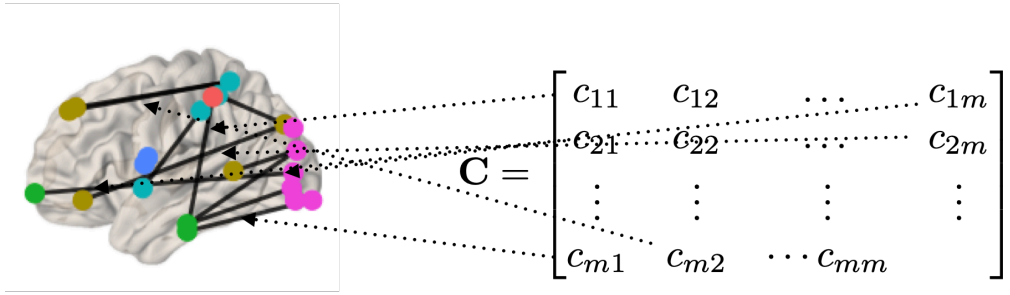


VNNs provide an anatomically interpretable and explainable brain age gap

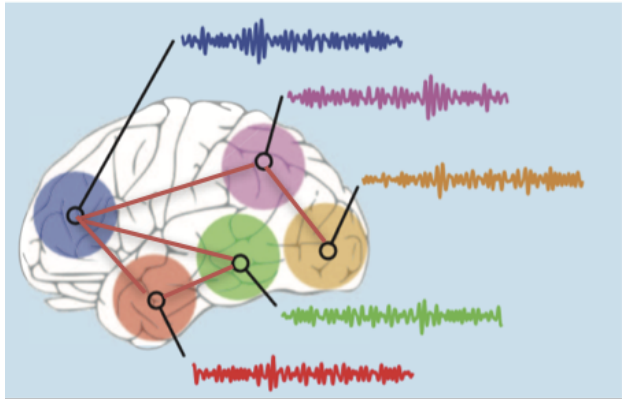


Network neuroscience

- Modeling brain as a network (**connectomes**)



Anatomical covariance matrix
(structural connectome)

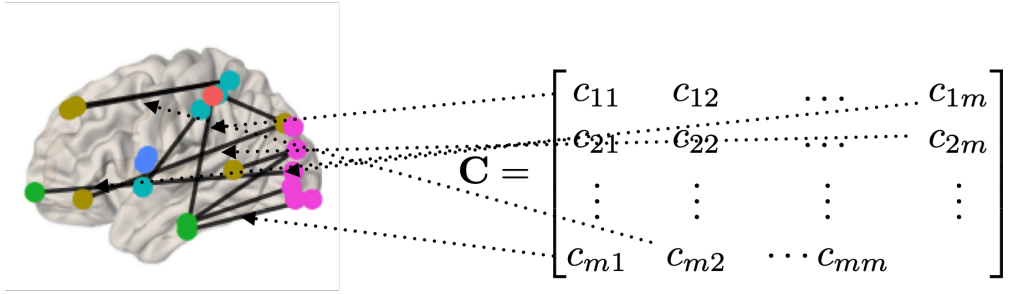


Functional connectome

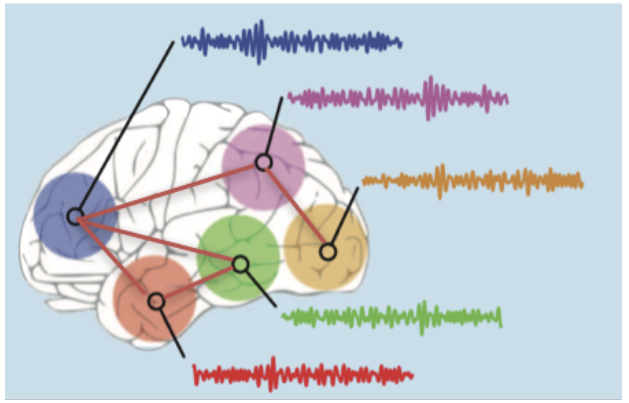
- **Motivation**
  - Significant redundancies in brain structural/functional features
  - Brain structure/function is compromised in neurodegeneration

Covariance matrices in network neuroscience

- Covariance matrices appear commonly in network neuroscience



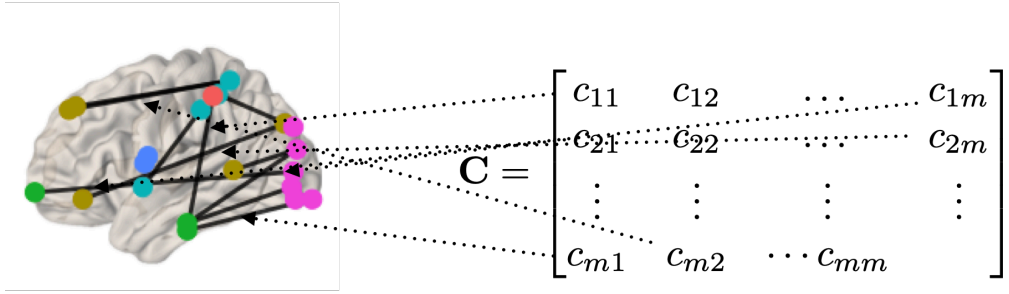
Anatomical covariance matrix



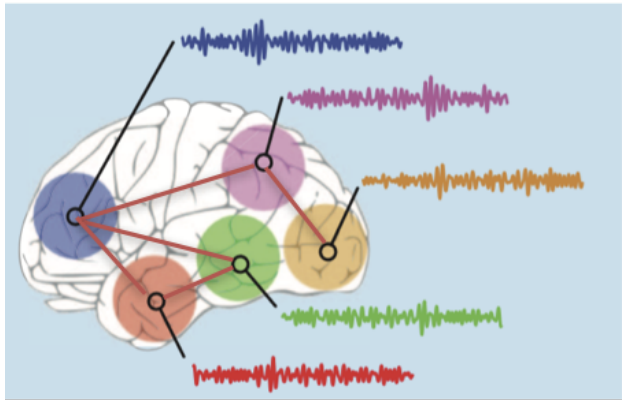
Functional connectome

Covariance matrices in network neuroscience

- Covariance matrices appear commonly in network neuroscience



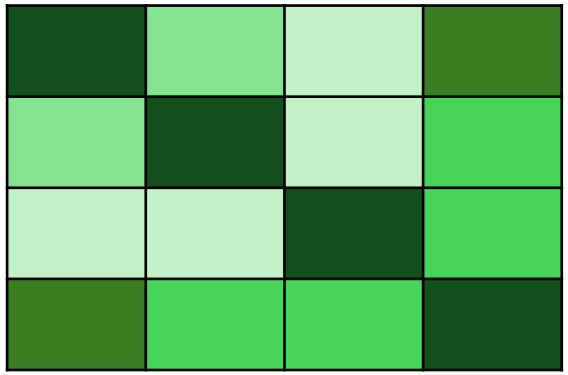
Anatomical covariance matrix



Functional connectome

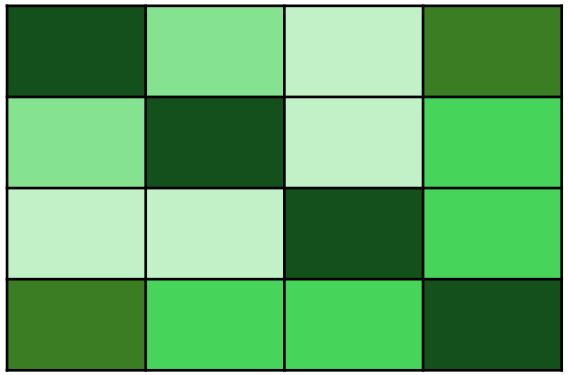
- Inference over covariance matrices in ML

  - **Traditional** statistical approaches (for e.g., PCA)
    - Interpretable, suitable for low data regimes
  - **Deep learning** approaches (for e.g., GNNs)
    - Enhanced expressivity, improved performance

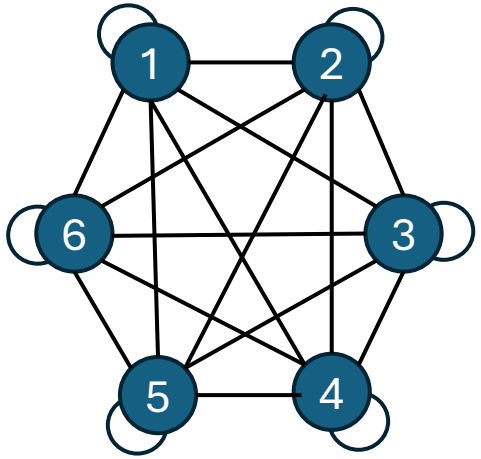


Covariance matrix as a graph

- Covariance matrix is a **data-driven** graph



$$\mathbf{x} = [x_1, \ldots, x_6]^\mathsf{T}$$

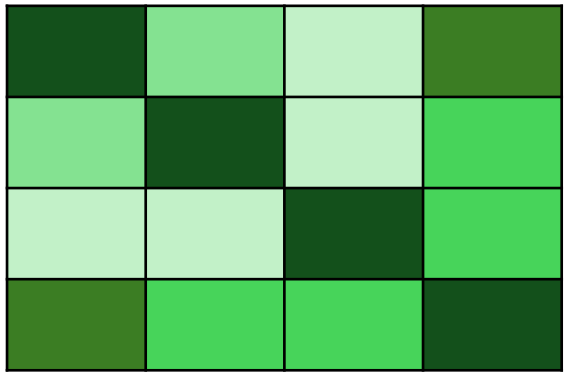


Covariance matrix as a fully-connected graph

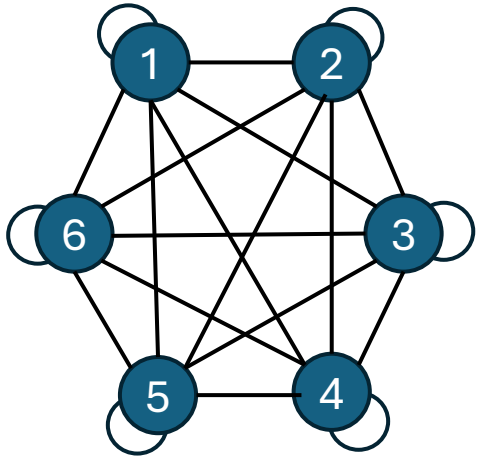
$$\hat{\mathbf{C}} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\mathsf{T}, \text{ where } \hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

Covariance matrix as a graph

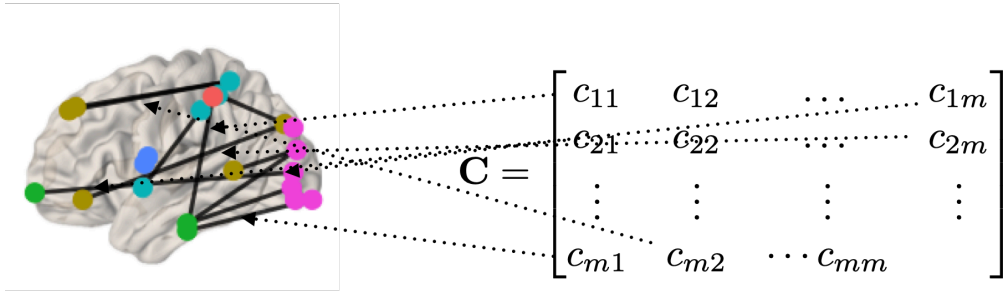
- Covariance matrix is a **data-driven** graph



$$\mathbf{x} = [x_1, \ldots, x_6]^\mathsf{T}$$



Covariance matrix as a fully-connected graph

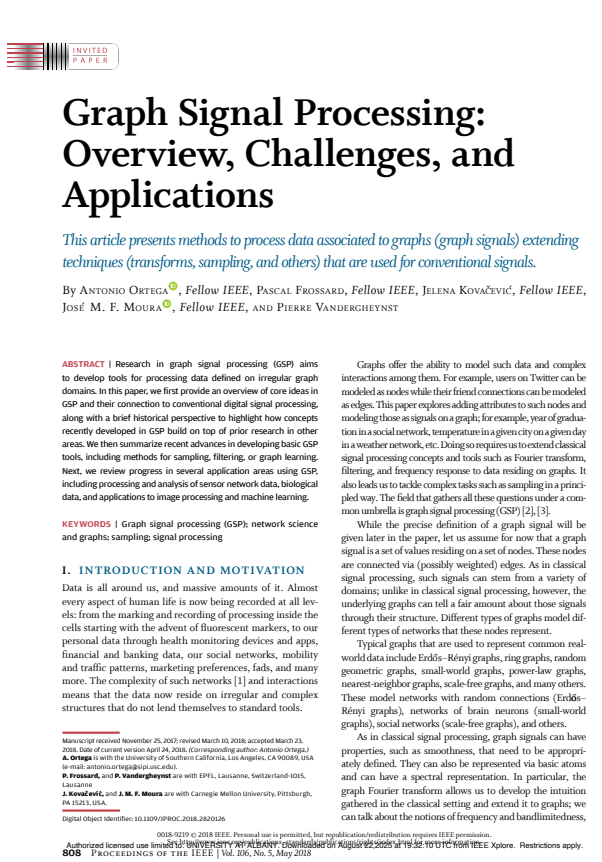


Anatomical covariance matrix
(estimated from cortical features)

$$\hat{\mathbf{C}} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\mathsf{T}, \text{ where } \hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

Graph signal processing

- Signal and information processing is about exploiting **signal structure**
- **Graph signal processing (GSP):** broaden classical signal processing to graphs



INVITED PAPER

Graph Signal Processing: Overview, Challenges, and Applications

This article presents methods to process data associated to graphs (graph signals) extending techniques (transforms, sampling, and others) that are used for conventional signals.

By Antonio Ortega, *Fellow IEEE*, Pascal Frossard, *Fellow IEEE*, Jelena Kovačević, *Fellow IEEE*, José M. F. Moura, *Fellow IEEE*, and Pierre Vandergheynst

ABSTRACT | Research in graph signal processing (GSP) aims to develop tools for processing data defined on irregular graph domains. In this paper, we first provide an overview of core ideas in GSP and their connection to conventional digital signal processing, along with a brief historical perspective to highlight how concepts recently developed in GSP build on top of prior research in other areas. We then summarize recent advances in developing basic GSP tools, including methods for sampling, filtering, or graph learning. Next, we review progress in several application areas using GSP, including processing and analysis of sensor network data, biological data, and applications to image processing and machine learning.

KEYWORDS | Graph signal processing (GSP); network science and graphs; sampling; signal processing

I. Introduction and Motivation

Data is all around us, and massive amounts of it. Almost every aspect of human life is now being recorded at all levels: from the marking and recording of processing inside the cells starting with the advent of fluorescent markers, to our personal data through health monitoring devices and apps, financial and banking data, our social networks, mobility and traffic patterns, marketing preferences, fads, and many more. The complexity of such networks [1] and interactions means that the data now reside on irregular and complex structures that do not lend themselves to standard tools.

Graphs offer the ability to model such data and complex interactions among them. For example, users on Twitter can be modeled as nodes while their friend connections can be modeled as edges. This paper explores adding attributes to such nodes and modeling those as signals on a graph; for example, year of graduation in a social network, temperature in a given city on a given day in a weather network, etc. Doing so requires us to extend classical signal processing concepts and tools such as Fourier transform, filtering, and frequency response to data residing on graphs. It also leads us to tackle complex tasks such as sampling in a principled way. The field that gathers all these questions under a common umbrella is graph signal processing (GSP) [2], [3].

While the precise definition of a graph signal will be given later in the paper, let us assume for now that a graph signal is a set of values residing on a set of nodes. These nodes are connected via (possibly weighted) edges. As in classical signal processing, such signals can stem from a variety of domains; unlike in classical signal processing, however, the underlying graphs can tell a fair amount about those signals through their structure. Different types of graphs model different types of networks that these nodes represent.

Typical graphs that are used to represent common real-world data include Erdős–Rényi graphs, ring graphs, random geometric graphs, small-world graphs, power-law graphs, nearest-neighbor graphs, scale-free graphs, and many others. These model networks with random connections (Erdős–Rényi graphs), networks of brain neurons (small-world graphs), social networks (scale-free graphs), and others.

As in classical signal processing, graph signals can have properties, such as smoothness, that need to be appropriately defined. They can also be represented via basic atoms and can have a spectral representation. In particular, the graph Fourier transform allows us to develop the intuition gathered in the classical setting and extend it to graphs; we can talk about the notions of frequency and bandlimitedness,

Manuscript received November 25, 2017; revised March 10, 2018; accepted March 23, 2018. Date of current version April 24, 2018. (*Corresponding author: Antonio Ortega.*)
A. Ortega is with the University of Southern California, Los Angeles, CA 90089, USA (e-mail: antonio.ortega@sipi.usc.edu).
P. Frossard, and **P. Vandergheynst** are with EPFL, Lausanne, Switzerland-1015, Lausanne.
J. Kovačević, and **J. M. F. Moura** are with Carnegie Mellon University, Pittsburgh, PA 15213, USA.
Digital Object Identifier: 10.1109/JPROC.2018.2820126

75TH ANNIVERSARY OF SIGNAL PROCESSING SOCIETY SPECIAL ISSUE

Geert Leus, Antonio G. Marques, José M.F. Moura, Antonio Ortega, and David I Shuman

Graph Signal Processing

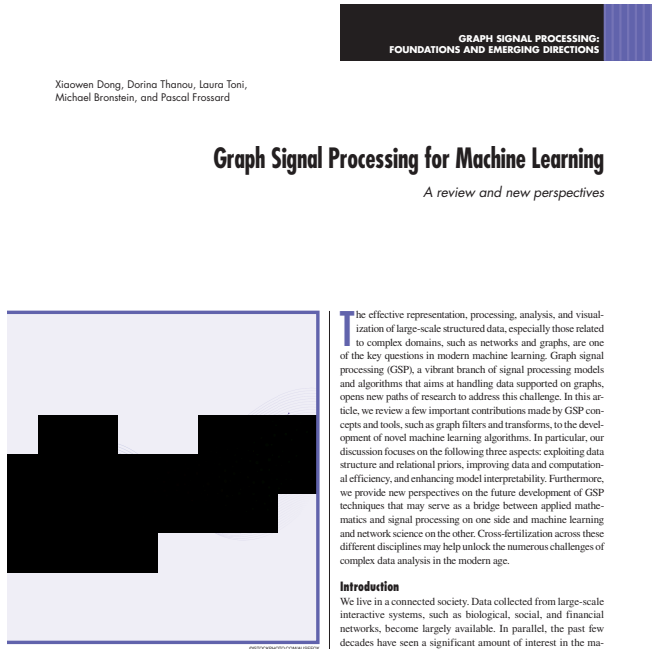
History, development, impact, and outlook

Digital Object Identifier 10.1109/MSP.2023.3262906
Date of current version: 1 June 2023

Signal processing (SP) excels at analyzing, processing, and inferring information defined over regular (first continuous, later discrete) domains such as time or space. Indeed, the last 75 years have shown how SP has made an impact in areas such as communications, acoustics, sensing, image processing, and control, to name a few. With the digitalization of the modern world and the increasing pervasiveness of data-collection mechanisms, information of interest in current applications oftentimes arises in non-Euclidean, irregular domains. Graph SP (GSP) generalizes SP tasks to signals living on non-Euclidean domains whose structure can be captured by a weighted graph. Graphs are versatile, able to model irregular interactions, easy to interpret, and endowed with a corpus of mathematical results, rendering them natural candidates to serve as the basis for a theory of processing signals in more irregular domains.

The term *graph signal processing* was coined a decade ago in the seminal works of [1], [2], [3], and [4]. Since these papers were published, GSP-related problems have drawn significant attention, not only within the SP community [5] but also in machine learning (ML) venues, where research in graph-based learning has increased significantly [6]. Graph signals are well-suited to model measurements/information/data associated with (indexed by) a set where 1) the elements of the set belong to the same class (regions of the cerebral cortex, members of a social network, weather stations across a continent); 2) there exists a relation (physical or functional) of proximity, influence, or association among the different elements of that set; and 3) the strength of such a relation among the pairs of elements is not homogeneous. In some scenarios, the supporting graph is a physical, technological, social, information, or biological network where the links can be explicitly observed. In many other cases, the graph is implicit, capturing some notion of dependence or similarity across nodes, and the links must be inferred from the data themselves. As a result, GSP is a broad framework that encompasses and extends classical SP methods, tools, and algorithms to application domains of the modern technological world, including social, transportation, communication,

1053-5888/23©2023IEEE



GRAPH SIGNAL PROCESSING: FOUNDATIONS AND EMERGING DIRECTIONS

Xiaowen Dong, Dorina Thanou, Laura Toni, Michael Bronstein, and Pascal Frossard

Graph Signal Processing for Machine Learning

A review and new perspectives

Digital Object Identifier 10.1109/MSP.2020.3014591
Date of current version: 28 October 2020

The effective representation, processing, analysis, and visualization of large-scale structured data, especially those related to complex domains, such as networks and graphs, are one of the key questions in modern machine learning. Graph signal processing (GSP), a vibrant branch of signal processing models and algorithms that aims at handling data supported on graphs, opens new paths of research to address this challenge. In this article, we review a few important contributions made by GSP concepts and tools, such as graph filters and transforms, to the development of novel machine learning algorithms. In particular, our discussion focuses on the following three aspects: exploiting data structure and relational priors, improving data and computational efficiency, and enhancing model interpretability. Furthermore, we provide new perspectives on the future development of GSP techniques that may serve as a bridge between applied mathematics and signal processing on one side and machine learning and network science on the other. Cross-fertilization across these different disciplines may help unlock the numerous challenges of complex data analysis in the modern age.

Introduction

We live in a connected society. Data collected from large-scale interactive systems, such as biological, social, and financial networks, become largely available. In parallel, the past few decades have seen a significant amount of interest in the machine learning community for network data processing and analysis. Networks have an intrinsic structure that conveys very specific properties to data, e.g., interdependencies between data entities in the form of pairwise relationships. These properties are traditionally captured by mathematical representations such as graphs.

In this context, new trends and challenges have been developing fast. Let us consider, for example, a network of protein–protein interactions and the expression level of individual genes at every point in time. Some typical tasks in network biology related to this type of data are 1) discovery of key genes (via protein grouping) affected by the infection and 2) prediction of how the host organism reacts (in terms of gene expression)

1053-5888/20©2020IEEE

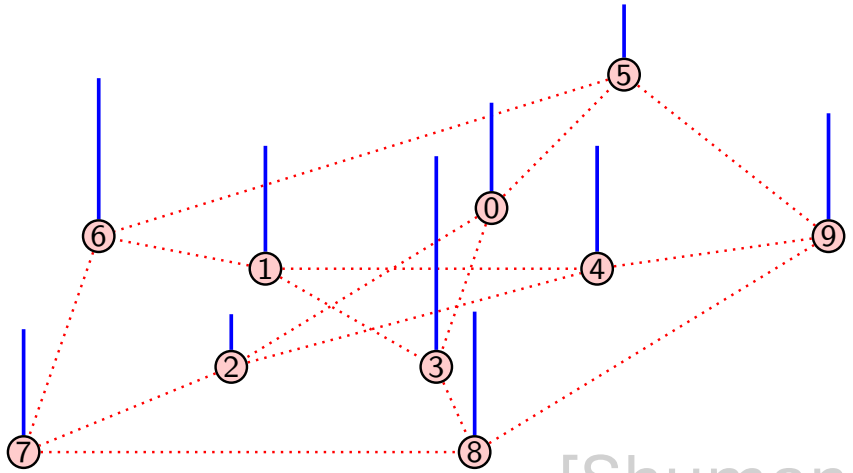
Graph signal

- **Graph signals** are mappings $x: V \mapsto \mathbb{R}$

  ⇨ graph signal is defined on the vertices of the graph

- **Graph signal** can be represented as a vector $\mathbf{x} \in \mathbb{R}^m$

  ⇨ x_i denotes the graph signal at i-th vertex in V



[Shuman, 2013]

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

Graph signal

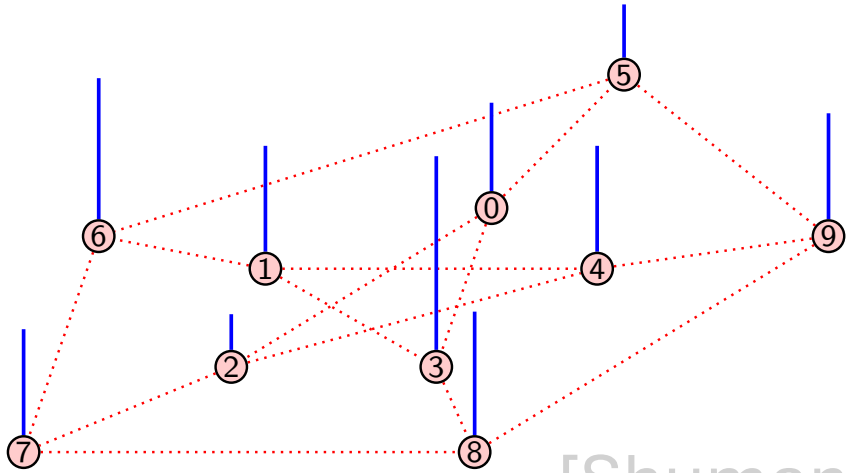
Cortical thickness features are graph signals

- **Graph signals** are mappings $x: V \mapsto \mathbb{R}$

  ⇨ graph signal is defined on the vertices of the graph

- **Graph signal** can be represented as a vector $\mathbf{x} \in \mathbb{R}^m$

  ⇨ x_i denotes the graph signal at i-th vertex in V



[Shuman, 2013]

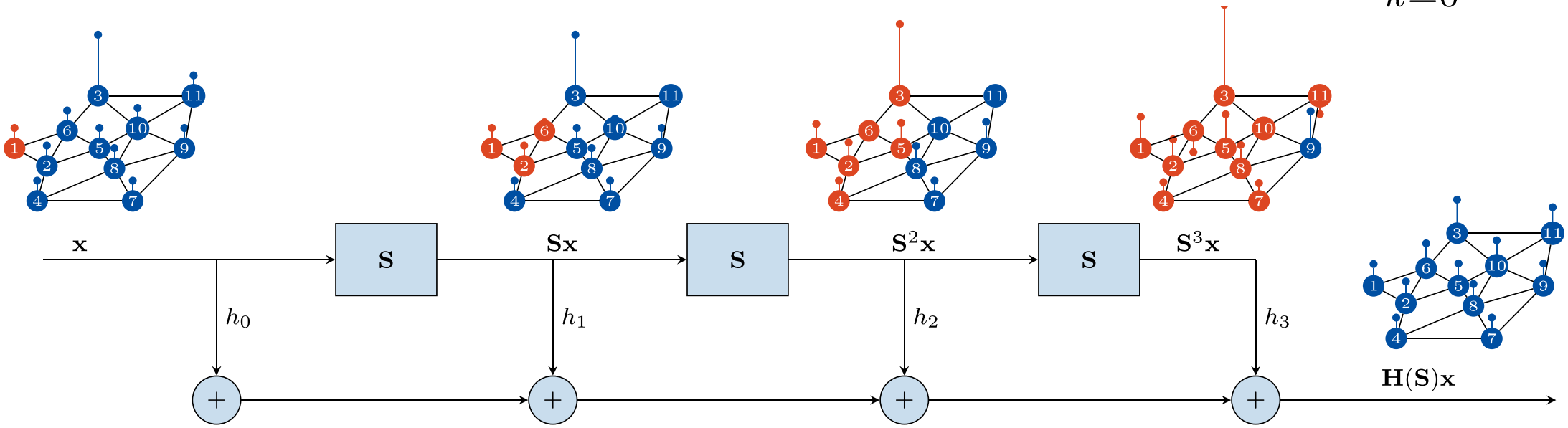
$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

Preliminaries: Graph filter

- **Graph filter $\mathbf{H}$** maps graph signal $\mathbf{x}$ to another graph signal $\mathbf{z}$ via linear-shift-and-sum operation

$$\mathbf{z} = \mathbf{H}(\mathbf{S})\mathbf{x},$$

$$\text{where } \mathbf{H} := h_0\mathbf{S}^0 + h_1\mathbf{S}^1 + h_2\mathbf{S}^2 + \cdots + h_K\mathbf{S}^K = \sum_{k=0}^{K} h_k\mathbf{S}^k$$

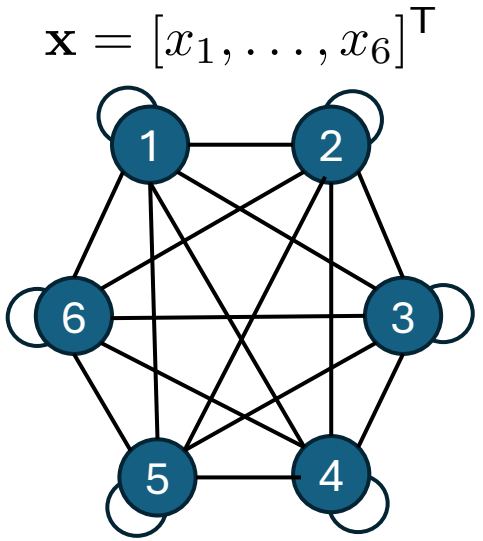


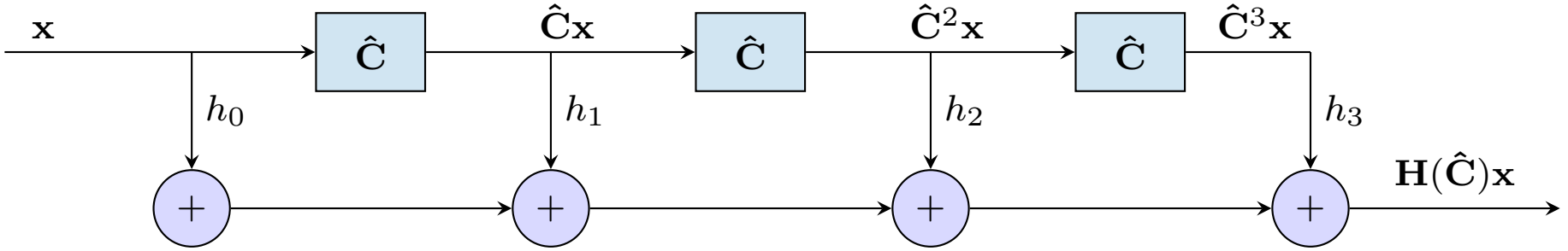
[Isufi et. al, IEEE TSP, 2024]

Graph filter on covariance matrix

- Covariance matrix forms a fully-connected graph where
  - nodes are features (brain regions)
  - edges are covariance values
- Graph filter on covariance matrix $\hat{\mathbf{C}}$ is defined as

$$\mathbf{H}(\hat{\mathbf{C}}) = \sum_{k=0}^{K} h_k \hat{\mathbf{C}}^k \mathbf{x}$$





CoVariance filter

- Analogy between $\mathbf{H}(\hat{\mathbf{C}})$ and PCA
  - Using eigendecomposition $\hat{\mathbf{C}} = \hat{\mathbf{V}}\hat{\mathbf{\Lambda}}\hat{\mathbf{V}}^{\mathsf{T}}$, it follows that

$$\mathbf{z} = \mathbf{H}(\hat{\mathbf{C}})\mathbf{x} = \sum_{k=0}^{K} h_k \hat{\mathbf{C}}^k \mathbf{x} = \sum_{k=0}^{K} h_k \hat{\mathbf{V}}\hat{\mathbf{\Lambda}}^k \hat{\mathbf{V}}^{\mathsf{T}}\mathbf{x} = \hat{\mathbf{V}}\Big(\underbrace{\sum_{k=0}^{K} h_k \hat{\mathbf{\Lambda}}^k}_{\text{Frequency response}}\Big)\underbrace{\hat{\mathbf{V}}^{\mathsf{T}}\mathbf{x}}_{\text{PCA}}$$

CoVariance filter

- Analogy between $\mathbf{H}(\hat{\mathbf{C}})$ and PCA
  - Using eigendecomposition $\hat{\mathbf{C}} = \hat{\mathbf{V}}\hat{\mathbf{\Lambda}}\hat{\mathbf{V}}^{\mathsf{T}}$, it follows that

$$\mathbf{z} = \mathbf{H}(\hat{\mathbf{C}})\mathbf{x} = \sum_{k=0}^{K} h_k \hat{\mathbf{C}}^k \mathbf{x} = \sum_{k=0}^{K} h_k \hat{\mathbf{V}}\hat{\mathbf{\Lambda}}^k \hat{\mathbf{V}}^{\mathsf{T}}\mathbf{x} = \hat{\mathbf{V}}\Big(\underbrace{\sum_{k=0}^{K} h_k \hat{\mathbf{\Lambda}}^k}_{\text{Frequency response}}\Big)\underbrace{\hat{\mathbf{V}}^{\mathsf{T}}\mathbf{x}}_{\text{PCA}}$$

  - coVariance filter and PCA are conceptually **equivalent**

$$\hat{\mathbf{V}}^{\mathsf{T}}\mathbf{z} = \Big(\sum_{k=0}^{K} h_k \hat{\mathbf{\Lambda}}^k\Big)\hat{\mathbf{V}}^{\mathsf{T}}\mathbf{x}$$

i-th component is modulated by $h(\lambda_i) = \sum_{k=0}^{K} h_k \lambda_i^k$

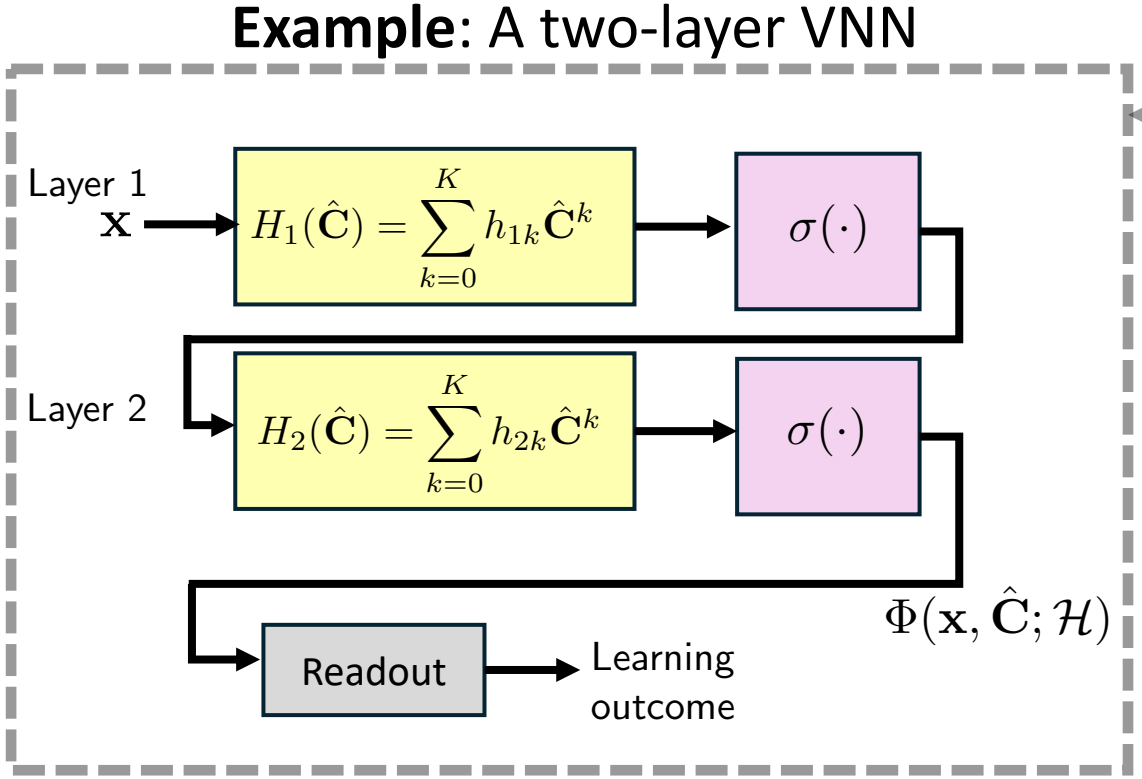
CoVariance Neural Networks (VNNs)

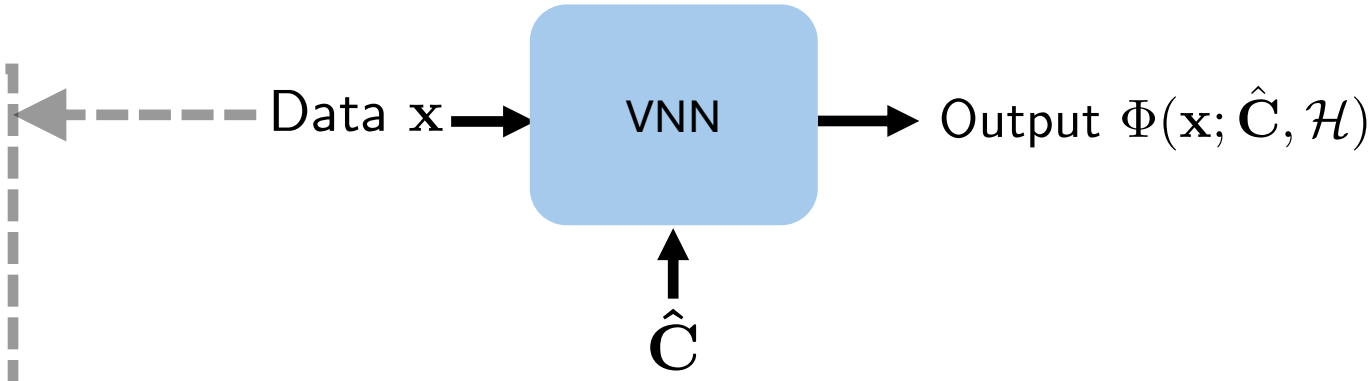
- coVariance filters can learn only **linear** representations
- To accommodate learn **non-linear** representations, concatenate coVariance filter with pointwise non-linearity σ (for e.g., ReLU, sigmoid, etc.)

CoVariance Neural Networks (VNNs)

- coVariance filters can learn only **linear** representations
- To accommodate learn **non-linear** representations, concatenate coVariance filter with pointwise non-linearity σ (for e.g., ReLU, sigmoid, etc.)

Example: A two-layer VNN

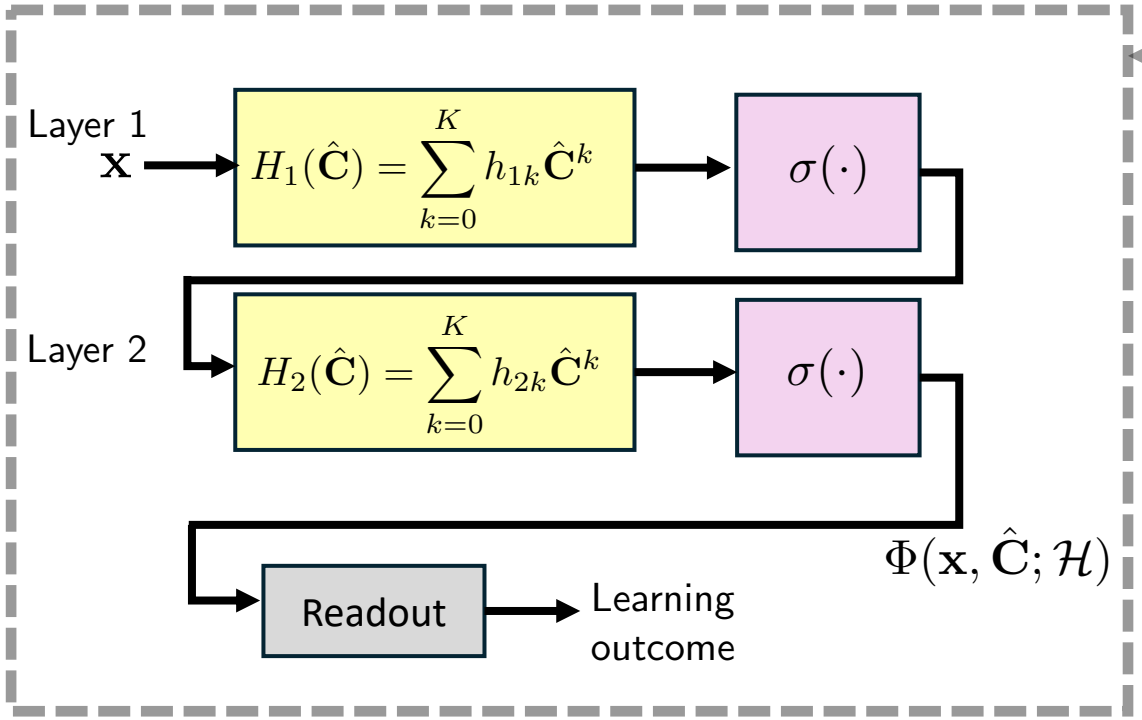


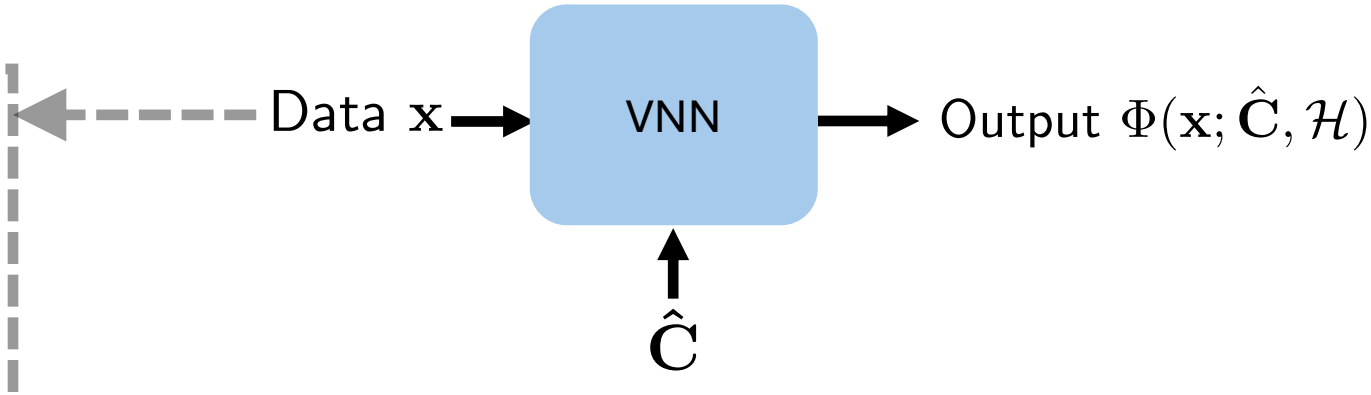


CoVariance Neural Networks (VNNs)

- coVariance filters can learn only **linear** representations
- To accommodate learn **non-linear** representations, concatenate coVariance filter with pointwise non-linearity σ (for e.g., ReLU, sigmoid, etc.)

Example: A two-layer VNN





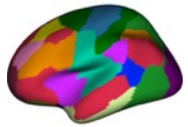
- $\Phi(\mathbf{x}; \hat{\mathbf{C}}, \mathcal{H})$ represents VNN output
- $\mathcal{H}$ is set of all filter taps

VNNs are well suited for neuroimaging data analysis

- Theoretical properties of VNNs make them appealing for neuroimaging data analysis
  - **Connections with PCA** ⇨ transparent outcomes by leveraging spectrum of covariance matrix
  - **Stability** ⇨ reproducible outcomes in limited data settings [Sihag et al., 2022]
  - **Transferability** ⇨ enhanced generalizability and robustness to choice of brain atlases [Sihag et al., 2024]

VNN vs PCA on age prediction task

- Regression task

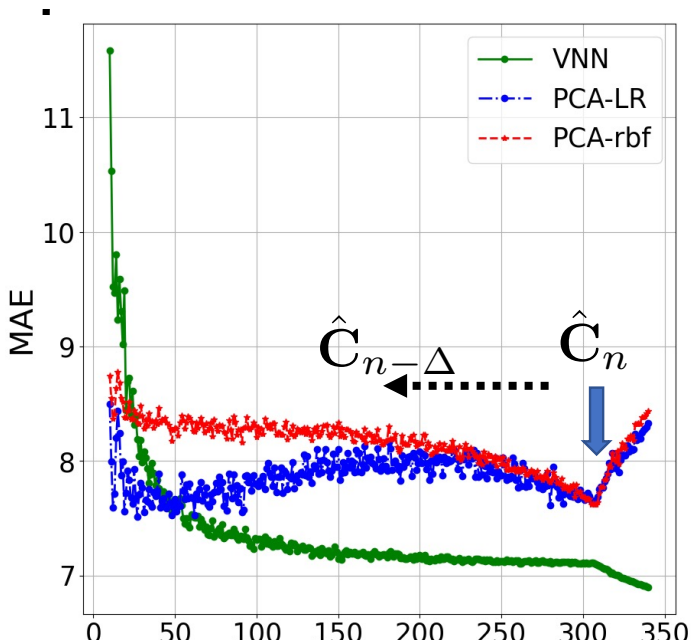


Cortical thickness data → VNN → Estimate of age

- Comparison against PCA-regression

  Data: cortical thickness dataset (m = 104) from (n = 341) human subjects

- **Metric**: MAE (mean absolute error)



VNN: coVariance Neural Network

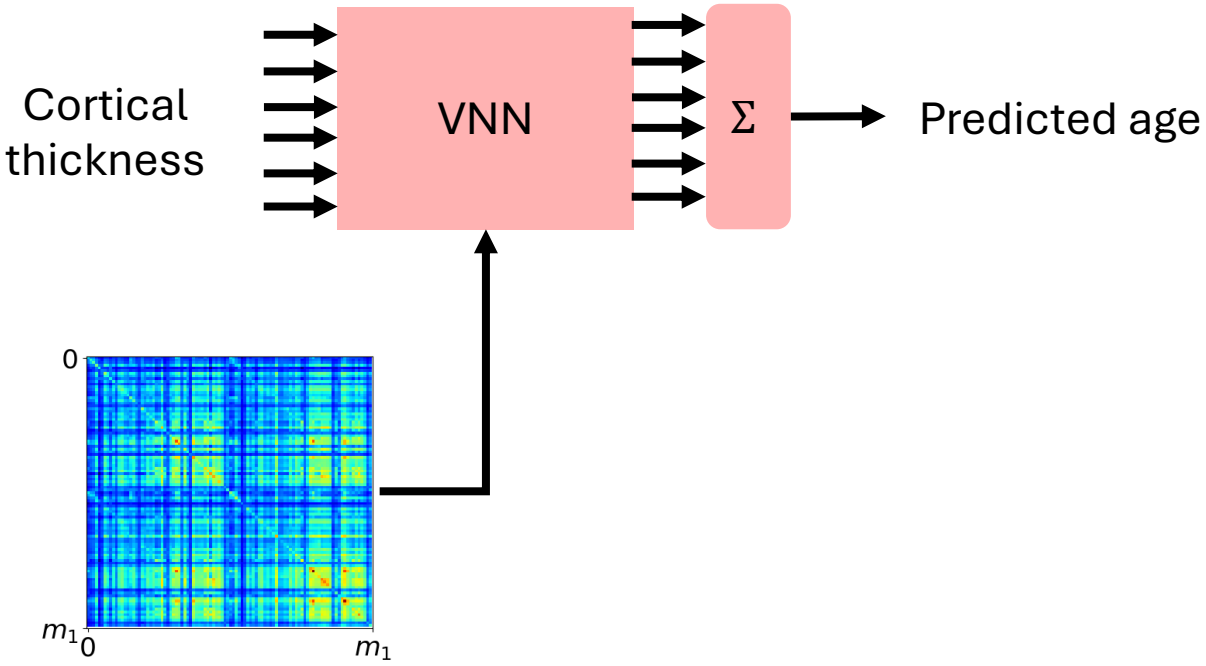
PCA-LR: PCA-regression with linear kernel

PCA-rbf: PCA regression with rbf kernel

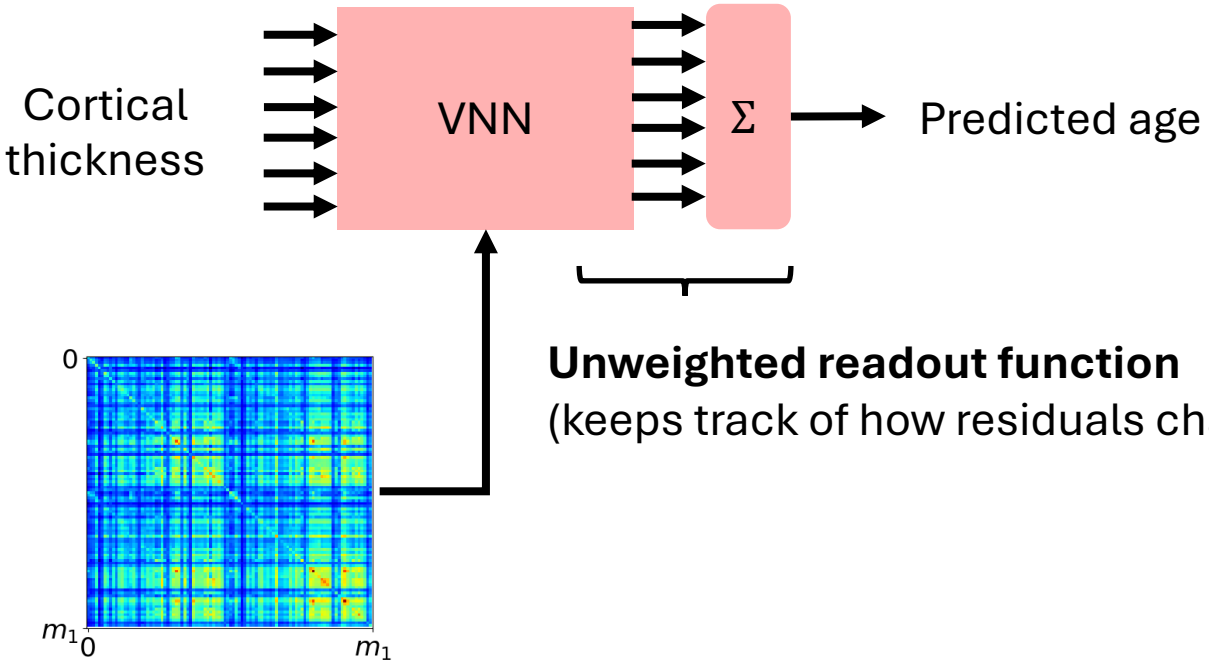
VNN outperforms PCA and is **more stable**

[Sihag et al., 2022]

VNNs provide an anatomically interpretable and explainable brain age gap

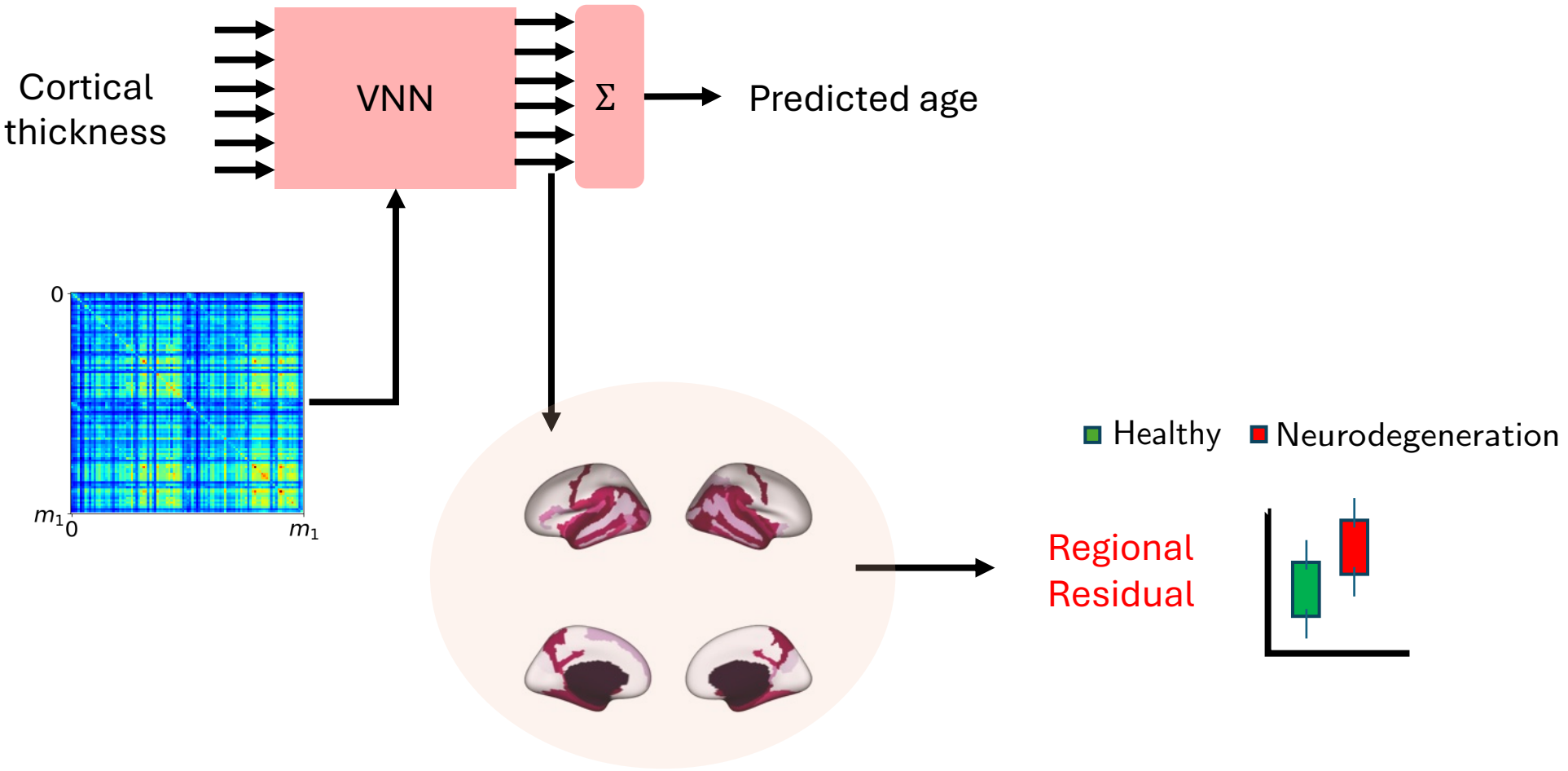


VNNs provide an anatomically interpretable and explainable brain age gap

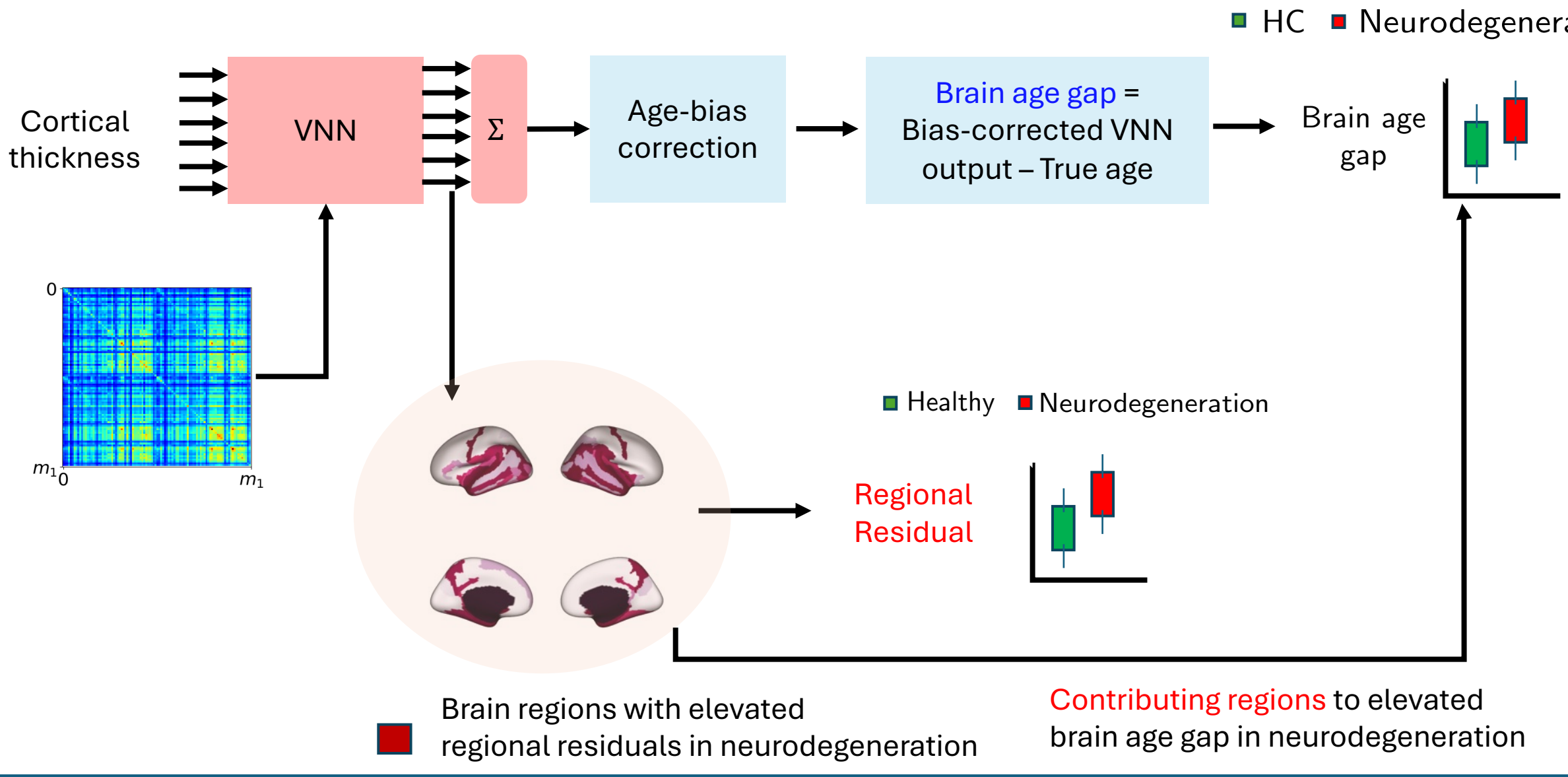


Unweighted readout function
(keeps track of how residuals change in target population)

VNNs provide an anatomically interpretable and explainable brain age gap



VNNs provide an anatomically interpretable and explainable brain age gap



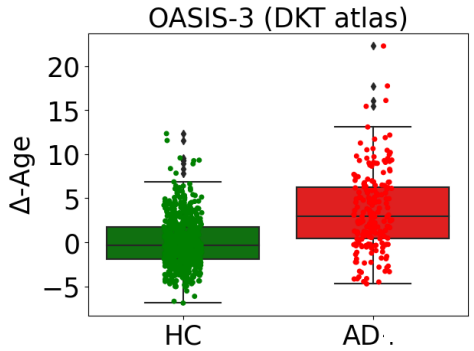
Experiments

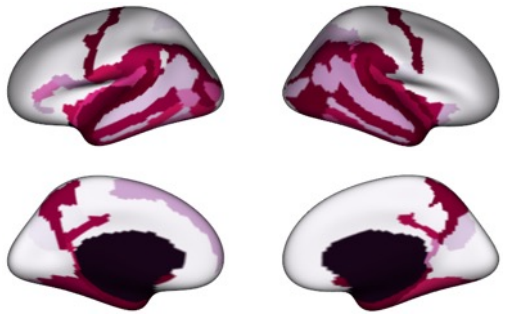
- Participants from OASIS-3 dataset [*], 148 cortical thickness features per individual (Distrieux brain atlas)

	HC	AD
Number	611	194
Age	68.38 (7.62)	74.72 (7.02)
Sex (m/f)	260/351	100/94
CDR sum of boxes	0	3.45 (1.74)

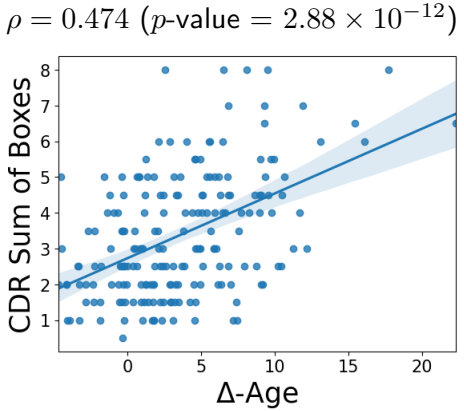
HC group: cognitively normal
AD group: AD diagnosis
CDR: Clinical dementia rating

- Brain age gap is elevated in AD group and correlated with CDR sum of boxes





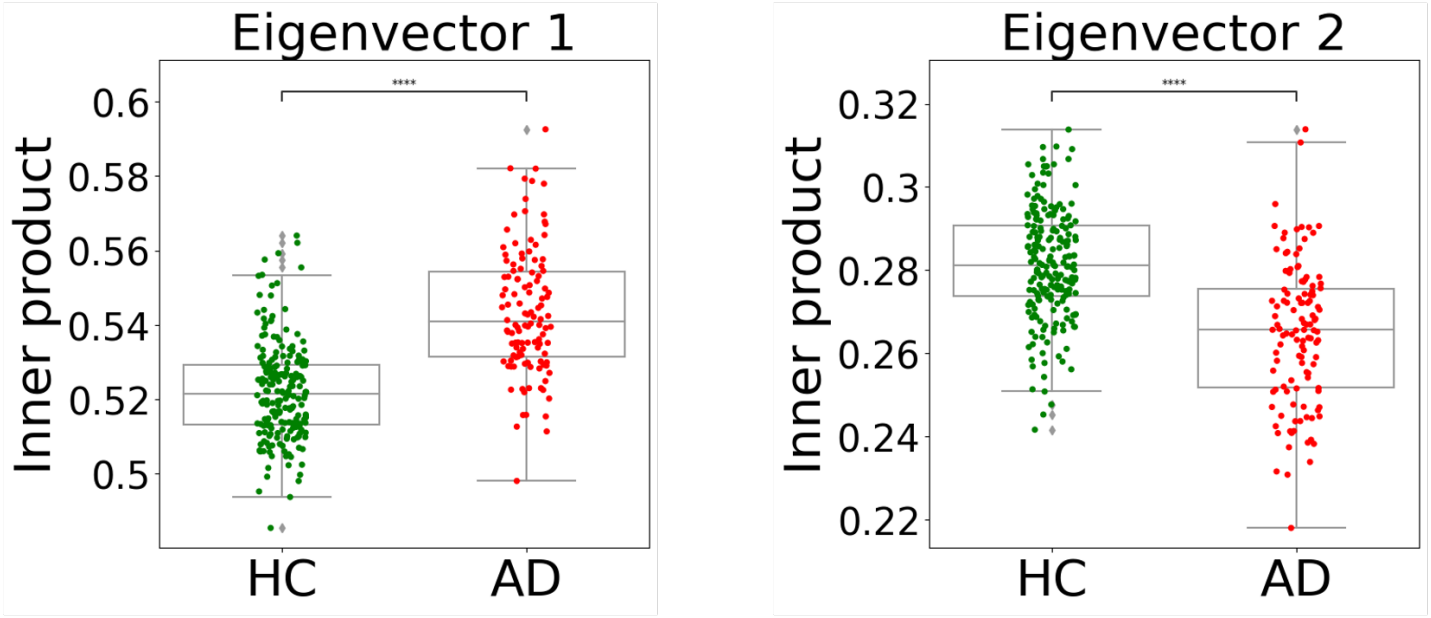
Anatomical interpretability



[*] Pamela J LaMontagne, et al. OASIS-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and Alzheimer disease. MedRxiv, 2019

Experiments

- VNN **distinctly** exploits eigenvectors in AD and HC groups



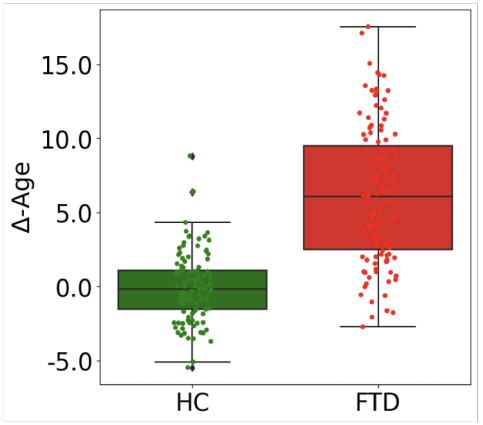
⇨ explains anatomical interpretability of brain age gap in AD

Experiments

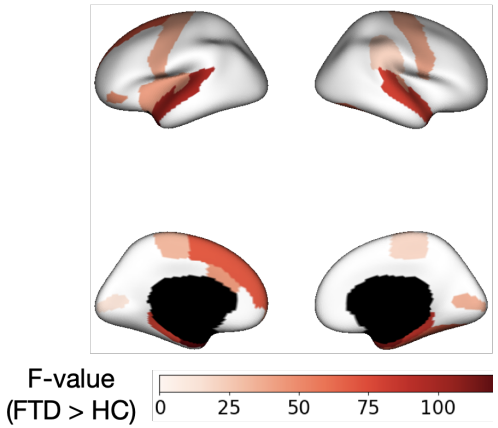
- Whole brain cortical thickness dataset for Frontotemporal Dementia (FTD)
  - Healthy controls (HC, n =114, age = 64.51 ± 6.51 years, 65 females)
  - FTD diagnosis (FTD, n =119, age = 64.72 ± 6.78 years, 47 females)

- 68 cortical thickness features (Desikan-Killiany atlas)

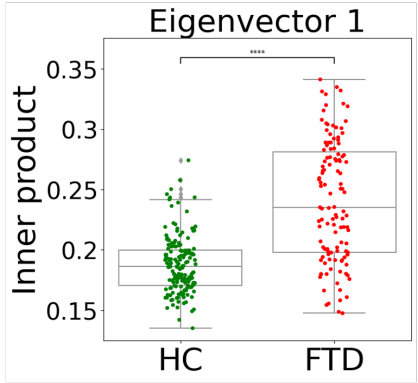
Brain age gap distributions

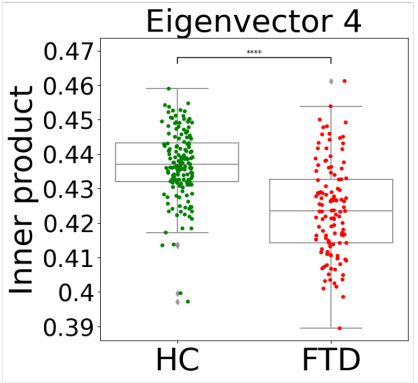


Anatomic interpretability



Explaining anatomic interpretability

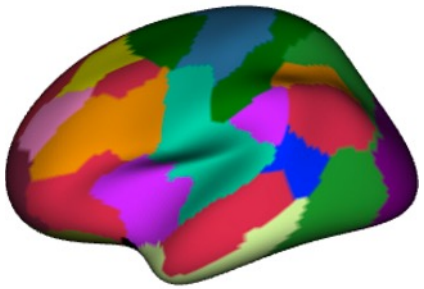


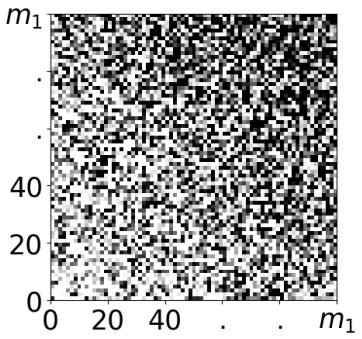


Brain age gap prediction on multi-scale datasets

- Datasets capture information about same phenomenon at **different scales**

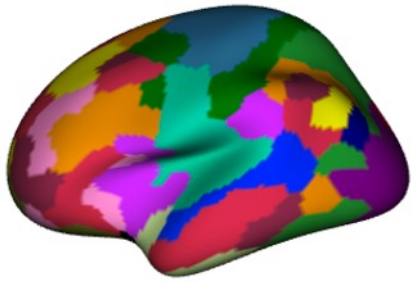
Dataset with m_1 features

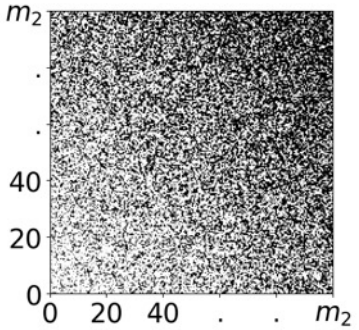




Covariance matrix $\mathbf{C}_{m_1}$
(size $m_1 \times m_1$)

Dataset with m_2 features

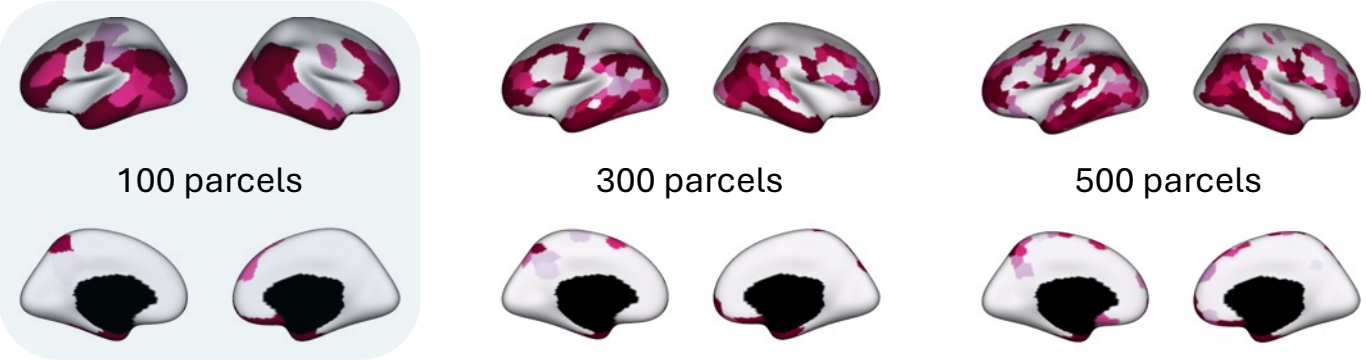


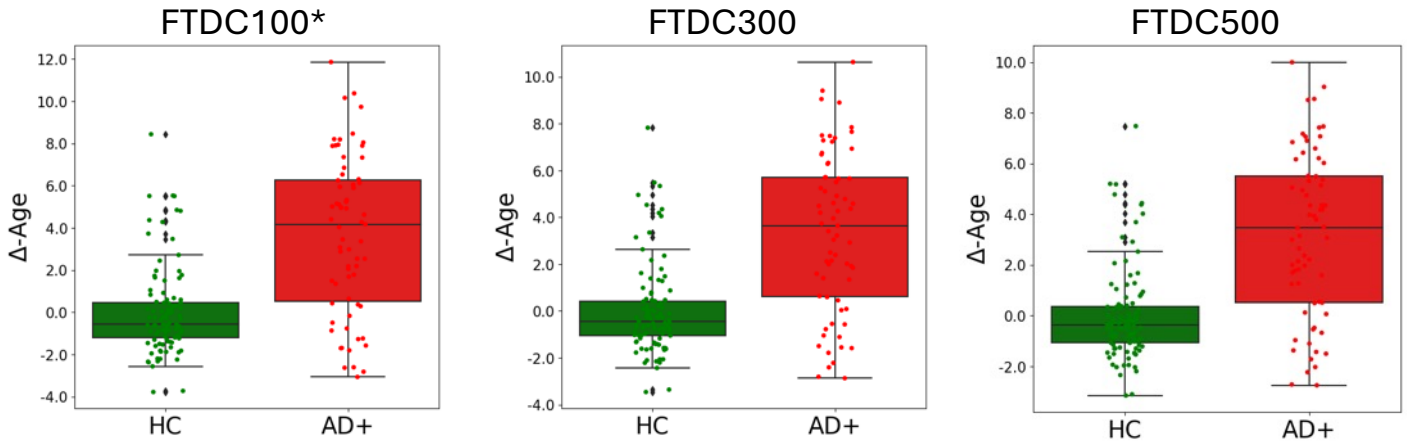


Covariance matrix $\mathbf{C}_{m_2}$
(size $m_2 \times m_2$)

Recap: Transferability of VNNs cross-validates brain age gap in multi-resolution setting

Objective: Brain age gap prediction in HC (healthy) and AD+ (Alzheimer's) cohorts from VNNs trained on 100-feature dataset

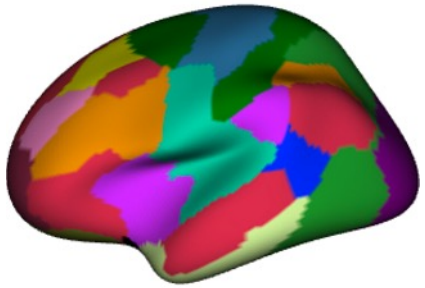




- ROIs contributing to elevated brain age gap in AD+ across different resolutions
- Brain age gap is elevated in AD+ w.r.t HC cohort in 100-feature dataset
- Results on brain age gap retained after transferring VNN to 300 and 500-feature datasets

Conclusions

- Brain age gap prediction models show wide generalizability



- VNNs provide a **principled** perspective to brain age gap
  - anatomically interpretable and explainable

- VNN-derived brain age is **a biomarker** for tracking neurodegeneration and disease monitoring

- **Transferability** of VNNs help cross-validate interpretability

References

Sihag, Saurabh, Mateos, Gonzalo, C. McMillan, and Ribeiro, Alejandro, “coVariance neural networks,” in Proc. Conference on Neural Information Processing Systems, Nov. 2022.

Saurabh Sihag, Gonzalo Mateos, C. McMillan, and Alejandro Ribeiro, “Explainable brain age prediction using covariance neural networks,” in Proc. Conference on Neural Information Processing Systems, 2023.

Saurabh Sihag, Gonzalo Mateos, and Alejandro Ribeiro, “Disentangling neurodegeneration with brain age gap prediction models: A graph signal processing perspective,” in IEEE Signal Processing Magazine, 2025 (to appear).

Sihag, Saurabh, Mateos, Gonzalo, C. McMillan, and Ribeiro, Alejandro, “Transferability of covariance neural networks,”, in IEEE Journal of Selected Topics in Signal Processing, pp. 1–16, 2024.

S. Khalafi, Saurabh Sihag, and Alejandro Ribeiro, “Neural tangent kernels motivate cross-covariance graphs in neural networks,” in International Conference on Machine Learning, 2024.

Sihag, Saurabh, Mateos, Gonzalo, and Ribeiro, Alejandro, “Explainable brain age gap prediction in neurode-generative conditions using covariance neural networks,” IEEE International Symposium on Biomedical Imaging, 2025.

A. Cavallo, Z. Gao, and Elvin Isufi, “Sparse covariance neural networks,” arXiv:2410.01669, vol. cs.LG, 2024.

Cavallo, Andrea, et al. "Fair covariance neural networks." IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2025.

A. Cavallo, M. Sabbaqi, and Isufi, Elvin, “Spatiotemporal covariance neural networks,” in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 18–34, Springer, 2024.