Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Predicting Cognitive Scores With Graph Neural Networks Through Sample Selection Learning

Martin Hanik
Zuse Institute Berlin
Mehmet Arif Demirtaş
Istanbul Technical University
Mohammed Amine Gharsallaoui
Istanbul Technical University
Islem Rekik (irekik@itu.edu.tr)
University of Dundee https://orcid.org/0000-0001-5595-6673

Research Article

Keywords: Regression, Graph Neural Network, Sample Selection, Functional Brain Connectome, Cognitive Score Prediction

Posted Date: July 6th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-634170/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Brain Imaging and Behavior manuscript No.
(will be inserted by the editor)

Predicting cognitive scores with graph neural networks through sample selection learning

Martin Hanik · Mehmet Arif Demirtaş · Mohammed Amine Gharsallaoui · Islem Rekik

Received: date / Accepted: date

Abstract Analyzing the relation between intelligence and neural activity is of the utmost importance in understanding the working principles of the human brain in health and disease. In existing literature, functional brain connectomes have been used successfully to predict cognitive measures such as intelligence quotient (IQ) scores in both healthy and disordered cohorts using machine learning models. However, existing methods resort to flattening the brain connectome (i.e., graph) through vectorization which overlooks its topological properties. To address this limitation and inspired from the emerging graph neural networks (GNNs), we design a novel regression GNN model (namely RegGNN) for predicting IQ scores from brain connectivity. On top of that, we introduce a novel, fully modular sample selection method to select the best samples to learn from for our target prediction task. However, since such deep learning architectures are computationally expensive to train, we further propose a *learning-based sample selection* method that learns how to choose the training samples with the highest expected predictive power on unseen samples. For this, we capitalize on the fact that connectomes (i.e., their adjacency matrices) lie in the symmetric positive definite (SPD) matrix cone. Our results on full-scale and verbal IQ prediction outperforms comparison methods in autism spectrum disorder cohorts and achieves a competitive performance for neurotypical subjects using 3-fold cross-validation. Furthermore, we show that our sample selection approach generalizes to other learning-based methods, which shows its usefulness beyond our GNN architecture.

Keywords Regression · Graph Neural Network · Sample Selection · Functional Brain Connectome · Cognitive Score Prediction

1 Introduction

Understanding how the structure of the brain influences cognitive scores such as IQ plays a vital role in understanding the working principles of the human brain. Cognitive scores are indicators of intellectual capacity which were found to be strongly connected to social factors: while high correlation between intelligence scores measured in childhood and educational success were observed in (Colom et al., 2007; Deary et al., 2007), they were also linked to health and mortality (Gottfredson and Deary, 2004; Batty et al., 2007). Motivated by this fact, many studies have investigated how far intelligence quotients (IQ) can be predicted from the structure of the brain. It was found, for example, that cerebral volume positively correlates with cognitive ability (Reiss et al., 1996; Mcdaniel, 2005). On a finer scale, activity and global connectivity of parts of the brain, especially

Martin Hanik and Mehmet Arif Demirtaş contributed equally to this work.

corresponding author: irekik@itu.edu.tr, http://basira-lab.com/

M. Hanik
Zuse Institute Berlin, Berlin, Germany

M. A. Demirtaş
BASIRA Lab, Faculty of Computer and Informatics, Istanbul Technical University, Istanbul, Turkey

M. A. Gharsallaoui
Ecole Polytechnique de Tunisie (EPT), Tunis, Tunisia; BASIRA Lab, Faculty of Computer and Informatics, Istanbul Technical University, Istanbul, Turkey

I. Rekik
BASIRA Lab, Faculty of Computer and Informatics, Istanbul Technical University, Istanbul, Turkey; Computing, School of Science and Engineering, University of Dundee, Dundee, UK

of the lateral prefrontal cortex, are linked to IQ (Gray et al., 2003; Woolgar et al., 2010; Cole et al., 2012, 2015).

Against this background, recent works have explored the possibility to predict cognitive ability scores from functional brain connectomes (Pamplona et al., 2015; Dubois et al., 2018; Dadi et al., 2019; Jiang et al., 2020; Dryburgh et al., 2020; He et al., 2020). Conventionally, connectomes are obtained from resting-state MRI and characterize the network structure of the brain; they are modeled as graphs whose nodes represent regions of interest (ROIs) and whose edges correspond to correlations in activity between these ROIs (Sporns et al., 2005). In order to achieve better generalizability across contexts and populations, (Shen et al., 2017) proposed a data-driven protocol for connectome-based predictive modeling of brain-behavior relationships, using cross-validation, to train a linear regression model. Building upon it, (Dryburgh et al., 2020) improved the results by evaluating negative and positive correlations of brain regions separately. They performed their analysis on both neurotypical subjects and subjects with Autism Spectrum Disorder (ASD) in order to investigate how neural correlates of intelligence scores are altered by atypical neurodevelopmental disorders.

Although such works achieved significant success, they mainly relied on classical machine learning approaches, which do not incorporate the *graph structure* of the connectomes; therefore, the local and global topological properties of the connectomes are not leveraged. (He et al., 2020) introduced graph neural networks (GNNs) (Wu et al., 2021), a subfield of *geometric deep learning*, where learning is customized to non-Euclidean spaces such as graphs with complex topologies (Dehmamy et al., 2019). GNNs are deep neural networks with graph convolution layers. They have already lead to significant increases in performance over existing methods in many fields. For example, they have been successfully applied to classification tasks on networks (Kipf and Welling, 2017; Qu et al., 2019), image segmentation (Qi et al., 2017), feature matching (Sarlin et al., 2020), few-shot learning (Garcia and Bruna, 2017; Kim et al., 2019), and various graph mining tasks (Schlichtkrull et al., 2018; Yun et al., 2019; Zhang et al., 2019). A very recent review on GNNs in the field of network neuroscience (Bessadok et al., 2021) examined a variety of graph-based architecture tailored for brain connectivity classification, integration, superresolution and synthesis across time and modalities. However, none of the reviewed methods were designed for brain graph regression for cognitive score prediction.

In this paper, we propose the first GNN architecture, namely *RegGNN*, that is specialized in regressing brain connectomes to a target cognitive score to predict. Our GNN utilizes graph convolutional layers to map input connectomes onto their corresponding cognitive scores, thereby allowing to extract the learned weights to identify the brain connectivities between anatomical regions that fingerprint the target score.

To improve the performance of the GNN, we additionally propose a novel *learning-based sample selection* method. It is independent from RegGNN and can be used with any architecture or regression learner. The method identifies training samples with the highest predictive power (i.e., those that are most likely to predict unseen test subjects with the lowest error); only these are then used for training. Through this, we eliminate the samples that do not increase—or even decrease—the prediction success of the model and reduce the computational resources needed for training the GNN.

Within our sample selection method, we make use of the fact that the (weighted) adjacency matrix of a functional brain connectome, when modeled as a correlation matrix, is symmetric positive semi-definite; and becomes symmetric positive definite after a simple regularization step (Dodero et al., 2015; Wong et al., 2018; You and Park, 2021). The space of SPD matrices forms a nonlinear manifold (Arsigny et al., 2006), and like (You and Park, 2021), we use a Riemannian geometric structure on it in order to obtain a *natural* notion of distance between two connectomes as well as tangent matrices that encode the paths that realize this distance.

We summarize the main contributions of our work as follows:

1. We introduce a novel, learning-based sample selection method for graph neural networks that helps to increase accuracy when predicting cognitive scores from connectomes.
2. We propose novel similarity measures between brain connectomes by combining notions from Riemannian geometry and topology of graphs. These measures can be used in other applications whenever we deal with objects that can be interpreted as elements of Riemannian manifolds.
3. We design a pipeline, consisting of RegGNN with sample selection, which outperforms state-of-the-art models in predicting full scale intelligence and verbal intelligence quotients from functional brain connectomes in an autism spectrum disorder cohort and achieves a competitive performance in a neurotypical cohort.

2 Methods

In this section, we detail the architecture of our RegGNN. Furthermore, we introduce our proposed sample selection method and show how we incorporate it into the training process of the GNN. To start with, we recount some facts on the Riemannian geometry of SPD matrices. Furthermore, we recall graph-topological centrality measures. The mathematical notations that we use in the following are summarized in Table 1.

2.1 Preliminaries

The space of n-by-n symmetric positive definite matrices $\mathrm{SPD}(n) = \{\mathbf{P} \in \mathbb{R}^{n,n} : \mathbf{P}^T = \mathbf{P}, \text{all eigenvalues of } \mathbf{P} \text{ are positive}\}$ forms a cone-like manifold in the set of all matrices of the same size (Faraut and Korányii, 1994). Being a manifold, there is a well-defined tangent space at every point $\mathbf{P} \in \mathrm{SPD}(n)$, which we denote by $T_{\mathbf{P}}\mathrm{SPD}(n)$. It is a basic fact that each $T_{\mathbf{P}}\mathrm{SPD}(n)$ can be identified with the set of symmetric n-by-n matrices. Therefore, in order to avoid later confusion, we call their elements *tangent matrices* instead of "tangent vectors" (which is the standard term in differential geometry).

As a manifold, $\mathrm{SPD}(n)$ can be endowed with a Riemannian geometric structure (do Carmo, 1992). Such a structure is determined by the choice of a Riemannian metric, i.e., a smoothly varying inner product on the tangent spaces. With its help, we can measure angles between (and norms of) tangent matrices. Furthermore, it induces a distance d on the space. Consequently, geodesics can be defined as (locally) shortest paths. Like a straight line in Euclidean space, a geodesic γ that connects two points $\mathbf{P}, \mathbf{Q} \in \mathrm{SPD}(n)$ can be represented by a unique tangent matrix $\log(\mathbf{P}, \mathbf{Q}) \in T_{\mathbf{P}}\mathrm{SPD}(n)$.[1] In particular, $\log(\mathbf{P}, \mathbf{Q})$ points in the direction of $\mathbf{Q}$, i.e., is parallel to γ at $\mathbf{P}$ and has norm (measured in the one induced from the metric) equal to the distance between $\mathbf{P}$ and $\mathbf{Q}$. Because of this, we can view $\log(\mathbf{P}, \mathbf{Q})$ as the linearized "difference" between $\mathbf{Q}$ and $\mathbf{P}$.

In contrast to Euclidean geometry, tangent matrices from different tangent spaces of a Riemannian manifold cannot be compared directly. Instead, they must be transported along curves to the same tangent space; this process is called *parallel translation*. This means that although tangent matrices at different points $\mathbf{P} \in \mathrm{SPD}(n)$ and $\mathbf{Q} \in \mathrm{SPD}(n)$ are symmetric matrices, we must bring them to a common point in order to compare them. The SPD space and parallel translation of vectors are illustrated in Fig. 1.

[1] It is denoted like this because the corresponding map is called *Riemannian logarithm*.

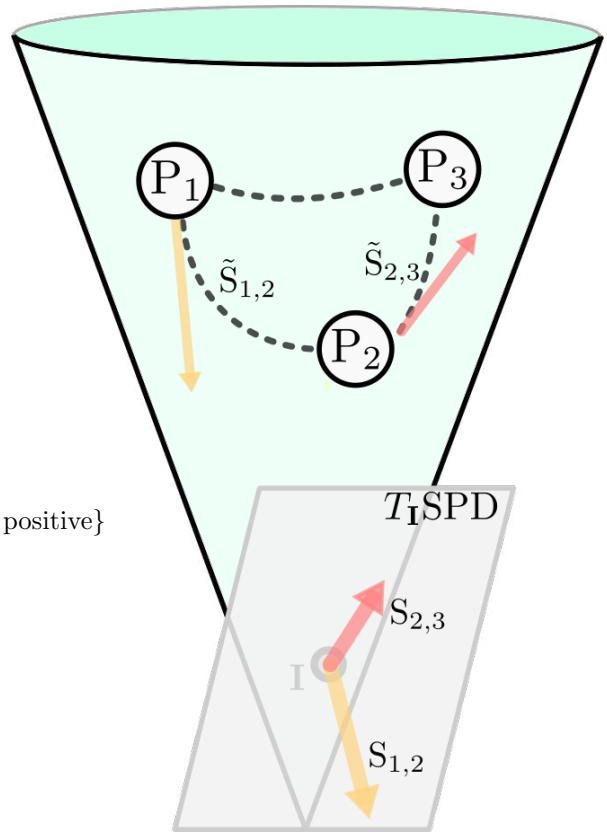

Fig. 1: Illustration of geodesics and parallel transport of tangent matrices on the SPD cone. The dashed lines are the geodesics between the matrices $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3 \in \mathrm{SPD}(n)$. The tangent matrices $\tilde{\mathbf{S}}_{1,2} := \log(\mathbf{P}_1, \mathbf{P}_2)$ and $\tilde{\mathbf{S}}_{2,3} := \log(\mathbf{P}_2, \mathbf{P}_3)$ are the yellow and red arrow, respectively; their parallel translations to $T_{\mathbf{I}}\mathrm{SPD}$ are $\mathbf{S}_{1,2}$ and $\mathbf{S}_{2,3}$.

Since all notions depend on the Riemannian structure, we must fix one. For $\mathrm{SPD}(n)$ several can be found in the literature, the most popular being the Log-Euclidean metric (Arsigny et al., 2006) and the affine-invariant metric (Moakher, 2005; Pennec et al., 2006). They have been applied successfully to connectomes for classification (Dodero et al., 2015; Yamin et al., 2020), regression (Wong et al., 2018), fingerprint extraction (Abbas et al., 2021), and statistical analysis (You and Park, 2021). We choose to work with the Log-Euclidean metric because it allows for comparatively efficient algorithms. Furthermore, parallel transport does not depend on the chosen path and a unique length-minimizing geodesic exists between any two points—both properties do not hold for most other metrics.

Notation	Dimension	Definition
n_{Train}	$\mathbb{N}$	number of subjects in the training group
n_{Test}	$\mathbb{N}$	number of subjects in the test group
d	$\mathbb{N}$	number of brain regions (i.e, ROIs)
n_s	$\mathbb{N}$	number of elements in the train-in group
n_h	$\mathbb{N}$	number of elements in the holdout group
$\text{SPD}(n)$	-	manifold of $n \times n$ symmetric positive definite matrices
$T_{\mathbf{P}}\text{SPD}(n)$	-	tangent space at $\mathbf{P} \in \text{SPD}(n)$ (can be identified with the vector space of symmetric matrices of the same size)
$\mathbf{I}$	$\mathbb{R}^{d\times d}$	identity matrix
$\mathbf{P}_i^s$	$\mathbb{R}^{d\times d}$	functional brain connectome (SPD) of subject i from the train-in group s
$\mathbf{P}_l^h$	$\mathbb{R}^{d\times d}$	functional brain connectome (SPD) of subject l from the holdout group h
$\tilde{\mathbf{S}}_{i,j}^{s,s}$	$\mathbb{R}^{d\times d}$	tangent matrix (symmetric matrix) encoding the geodesic between connectomes i and j from the train-in group s at $\mathbf{P}_i^s$
$\tilde{\mathbf{S}}_{j,l}^{s,h}$	$\mathbb{R}^{d\times d}$	tangent matrix (symmetric matrix) encoding the geodesic between connectomes j from the train-in group s and l from the holdout group h at $\mathbf{P}_j^s$
$\mathbf{S}_{i,j}^{s,s}$	$\mathbb{R}^{d\times d}$	parallel translation (symmetric matrix) of $\tilde{\mathbf{S}}_{i,j}^{s,s}$ to $T_{\mathbf{I}}\text{SPD}(n)$
$\mathbf{S}_{j,l}^{s,h}$	$\mathbb{R}^{d\times d}$	parallel translation (symmetric matrix) of $\tilde{\mathbf{S}}_{j,l}^{s,h}$ to $T_{\mathbf{I}}\text{SPD}(n)$
$v_{i,j}^{s,s}$	$\mathbb{R}^d$	feature vector extracted from $\mathbf{S}_{i,j}^{s,s}$
$v_{j,l}^{s,h}$	$\mathbb{R}^d$	feature vector extracted from $\mathbf{S}_{j,l}^{s,h}$
IQ_i^s	$\mathbb{R}$	cognitive score (IQ) of subject i from the train-in group s
IQ_l^h	$\mathbb{R}$	cognitive score (IQ) of subject l from the holdout group h

Table 1: Major mathematical notations used in this paper.

We now recall three basic topological centrality measures for an undirected[2] graph G: degree centrality, eigenvector centrality, and closeness centrality. They measure in how far a node is central to the (graph) network in the sense that most of the communication passes through it. A good reference on this is the book of (Fornito et al., 2016).

Let $\mathbf{A}$ be the (weighted) adjacency matrix of G, V the set of vertices of G, and $v \in V$.[3] The *degree centrality* $D(v)$ *of* v is defined by

$$D(v) := \sum_{\substack{w\in V\\ w\neq v}} \mathbf{A}_{vw}, \tag{1}$$

i.e., it assigns to each node its weighted sum of neighbors.

Let x be the unit norm eigenvector of A that corresponds to the largest eigenvalue λ_1 and has only non-negative entries. The *eigenvector centrality* $E(v)$ *of* v is the v-th entry of x; that is,

$$E(v) := \frac{1}{\lambda_1} \sum_{w\in V} \mathbf{A}_{vw} x_w, \tag{2}$$
$$\text{s.t. } \|x\|_2 = 1 \text{ and } x_w \geq 0 \text{ for all } w \in V.$$

It measures, in a relative sense, how influential a node is in the network. Intuitively, a high score means that a node has many neighbors that themselves have high eigenvector centrality scores.

[2] Of course, the centrality measures can also be defined for directed graphs but we do not need this here.

[3] With a slight abuse of notation, we identify nodes $v \in V$ and the integers we assign to them in order to construct A, e.g., we write A_{vw} for the entry that corresponds to the edge between nodes v and w.

Let l_{vw} be the length of the shortest path between two nodes v and w, and $n = |V|$. The *closeness centrality* $C(v)$ *of* v is defined by

$$C(v) := \frac{n-1}{\sum_{\substack{w\in V\\ w\neq v}} l_{vw}}, \tag{3}$$

i.e., as the inverse of the average distance of v to all other nodes.

We are now ready to introduce the graph neural network, and afterwards, the sample selection process.

2.2 RegGNN

Our GNN for regression, *RegGNN*, consists of two graph convolution layers and a downstream fully connected layer; a visualization is on the bottom left of Figure 2. In the following we denote the number of ROIs by d. Since adjacency matrices of connectomes are d-by-d correlation matrices $\mathbf{C}$, they can have zero (but no negative) eigenvalues. Therefore, we can simply regularize them to being symmetric positive definite by adding a small multiple of the identity matrix $\mathbf{I}$, i.e.,

$$\mathbf{P} := \mathbf{C} + \mu\mathbf{I} \tag{4}$$

for some small $\mu > 0$; see (Dodero et al., 2015) or (Wong et al., 2018). RegGNN receives the regularized adjacency matrix $\mathbf{P}$ of a connectome and predicts the corresponding IQ score from it by applying graph convolutions.

In the literature, there are various implementations of graph convolutions, which mainly differ by the propagation rule. Let $\mathbf{H}^{(i)}$ denote the activation matrix at

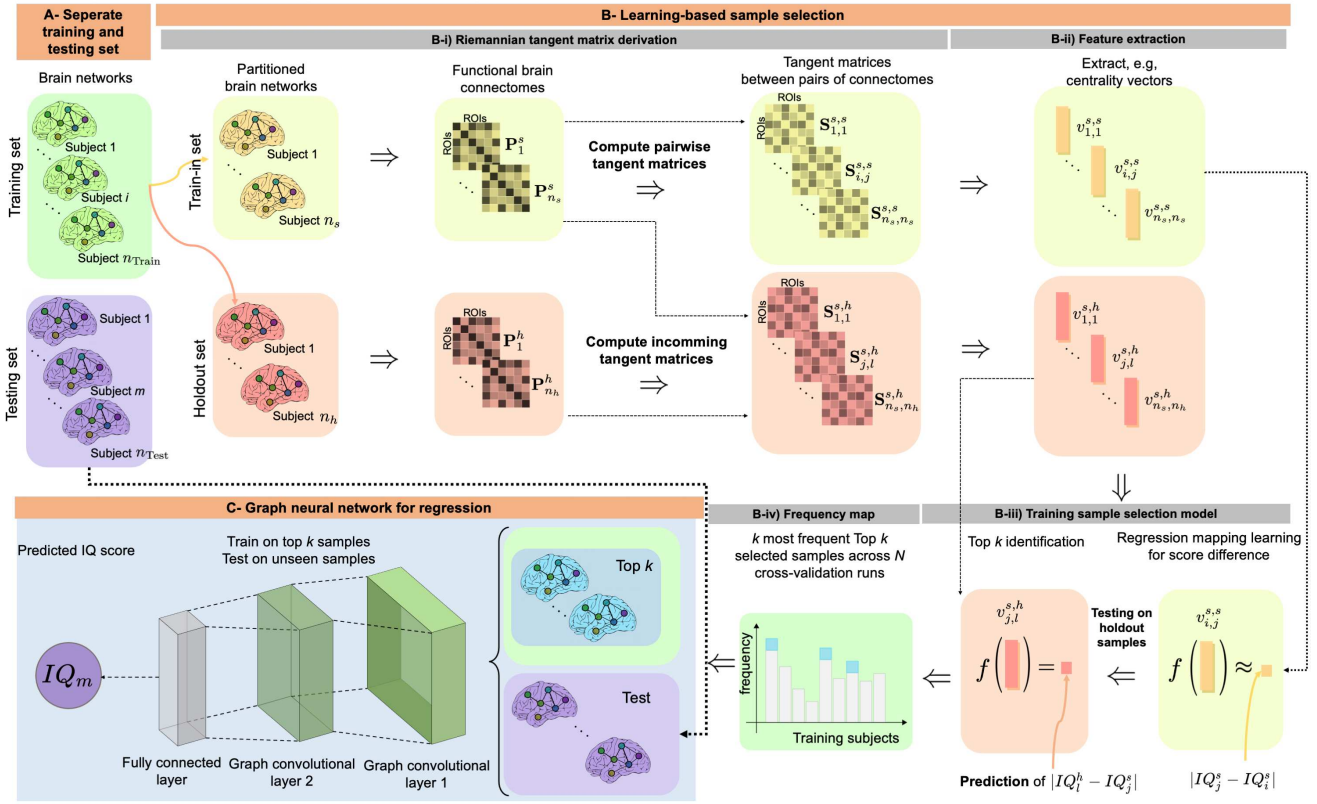

Fig. 2: *Illustration of the proposed sample selection strategy to train our regression graph neural network RegGNN.* **A)** We split data in training (green) and testing (violet) sets. **B-i)** We divide the training set into a train-in (yellow) and a holdout (red) set and extract tangent matrices for geodesics connecting elements from train-in set (yellow) and tangent matrices encoding geodesics from elements of the train-in to elements of the holdout group (red). **B-ii)** We compress the information from the tangent matrices through topological feature extraction into vectors. **B-iii)** We train linear regression mapping f on train-in-to-train-in feature vectors (yellow) to learn differences in target score and record for each element j of the train-group the k elements from the holdout group for whom the prediction difference in target score to j was smallest. **B-iv)** For each sample in the holdout set, we count how often it was among the top k predictors for a sample from the train-in group. **C)** After repeating B) in an N-fold cross validation manner, we select the k samples (blue) with the highest accumulated top-k frequency and train the graph neural network (only) on them. Finally, we use the testing set to evaluate the overall performance.

the i-th layer for $i = 0, 1, 2$. It is propagated to the next layer according to the general rule $\mathbf{H}^{(i+1)} = g_i(\mathbf{H}^{(i)}, \mathbf{P})$ with functions $g_i : \mathbb{R}^{d,d_{i-1}} \times \mathrm{SPD}(d) \to \mathbb{R}^{d,d_i}$ for $i = 1, 2$. As initialization we choose $\mathbf{H}^{(0)} := \mathbf{I}$. Furthermore, we choose the g_i as proposed by (Kipf and Welling, 2017). Define $\tilde{\mathbf{P}} := \mathbf{P} + \mathbf{I}$ and let $\tilde{\mathbf{D}}$ be the diagonal degree matrix of $\tilde{\mathbf{P}}$, we then formalize g_i as follows:

$$g_i(\mathbf{H}^{(i)}, \mathbf{P}) := \mathrm{ReLU}(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{P}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(i)}\mathbf{W}^{(i)}),$$

where $\mathbf{W}^{(i)} \in \mathbb{R}^{d,d_i}$ is the learnable weight matrix; we chose $d_0 := d = 116$, $d_1 := 64$, and $d_2 := 1$. We thus use the graph convolution layers to reduce the size of the connectomes and obtain an embedding for the brain graphs into $\mathbb{R}^d$. We apply a dropout layer after the first graph convolution operation for regularization. Finally, the obtained embedding passes through a fully connected layer (linear layer) which produces a continuous scalar output. The goal of the linear layer is to embed the resulting vector containing d features into a scalar value presenting the predicted IQ score.

2.3 Learning-based Sample Selection

We now introduce our *learning-based* sample selection strategy. The underlying idea is the following. Imagine

the (rather extreme) case that our subjects are clustered (possibly with outliers) in k tight groups according to their cognitive scores. Then, training a GNN on k representatives—one from each cluster—should yield good results; it should even perform better than a GNN trained on the full data set because it was not "distracted" by outliers during training. Ideally, as representatives we would choose the k most central samples of each group, i.e., those with the smallest average *difference in cognitive score* to the other samples. Now, since we want to predict cognitive scores from connectomes, we do not know the differences beforehand. On the other hand, if there is a relationship between the connectome of a subject and its cognitive score (which we always assume), we can use it to infer them. Therefore, our idea is to use the *differences* between the connectomes to learn the *differences* between the target scores in order to identify those "representatives". Our experiments below show that this idea of identifying few representatives and training only with them generalizes well to real data.

Implementing the idea, we represent differences between connectomes by tangent matrices and assume that the difference in IQ between two subjects depends linearly on (notions deduced from) the tangent matrix $\log(\mathbf{P}, \mathbf{Q})$ that encodes the geodesic between the corresponding connectomes $\mathbf{P}, \mathbf{Q} \in \mathrm{SPD}(d)$. This model is flexible, but at the same time allows for fast computations. Our sample selection method *learns* this linear map, which we call f in the following, via regression and uses it to identify the k samples with the lowest predicted average difference in target score to all other samples. As motivated above, we assume that they are representative of the whole set but do not contain (most of) the outliers that hinder successful training of the GNN. The structure and terminology of our method are inspired by the work of (Errica et al., 2019).

The sample selection method consists of four steps, which are visualized in part B of Fig. 2. Given a connectome data set, these are repeated in a nested N-fold cross-validation manner to make our selection of samples more robust. In cross-validation, we split the data set into two groups: a training subset which we call *train-in group*, and a validation subset which we call *holdout group*; we perform different train-in and holdout group splits so that each sample from the training set will be in the train-in group exactly $N-1$ times. We denote the (constant) sizes of the train-in and the holdout sets by n_s and n_h, respectively.

i) Riemannian tangent matrix derivation. For each pair of regularized connectomes $\mathbf{P}_i^s$, $\mathbf{P}_j^s \in \mathrm{SPD}(d)$ in the train-in group, we compute the tangent matrix

$$\tilde{\mathbf{S}}_{i,j}^{s,s} := \log(\mathbf{P}_i^s, \mathbf{P}_j^s) \in T_{\mathbf{P}_i^s}\mathrm{SPD}(d)$$

that encodes the geodesic between them and parallel translate it to $T_{\mathbf{I}}\mathrm{SPD}(d)$; we denote the resulting symmetric d-by-d matrix by $\mathbf{S}_{i,j}^{s,s}$. As a result, we obtain a set of $n_s\,(n_s-1)/2$ tangent matrices in $T_{\mathbf{I}}\mathrm{SPD}(d)$ that represent the pairwise differences between the connectomes from the train-in group. Analogously, we get a tangent matrix $\mathbf{S}_{j,l}^{s,h} \in T_{\mathbf{I}}\mathrm{SPD}(d)$ for each pair with one sample $\mathbf{P}_i^s$ from the train-in and another sample $\mathbf{P}_l^h$ from the holdout group; this results in another set consisting of $n_s\,n_h$ tangent matrices. The latter are the *outgoing* "difference matrices" *from the train-in into the holdout set.*

ii) Topological feature extraction from tangent matrices. The tangent matrices are still rather high dimensional, which leads to long computation times. Thus, we suggest to extract topological features in order to encode the information in more compact form. We select degree, closeness, and eigenvector centrality as well as combinations of them as our candidates for feature extraction. Note that in our case a tangent matrix represents the "difference" between two connectomes. The above features thus encode information on linearized *changes* in node connectivity. To the best of our knowledge, this is the first time that these notions were used in conjunction. As a result, from all in-group tangent matrices $\mathbf{S}_{i,j}^{s,s}$ as well as outgoing tangent matrices $\mathbf{S}_{j,l}^{s,h}$ we obtain feature vectors $v_{i,j}^{s,s}$ and $v_{i,j}^{s,h}$, respectively.

iii) Learning a linear regression mapping for predictive sample selection. We *learn* the linear map f via regression by training to map the vectors $v_{i,j}^{s,s}$ corresponding to samples i and j from the train-in group to the *absolute difference in target score* $|IQ_j^s - IQ_i^s|$ between them. We then apply the learned linear regression mapping f to the vectors $v_{j,l}^{s,h}$ to predict the differences in target score between all samples j from the train-in and samples l from the holdout group.

iv) Frequency map. We record *for each* holdout sample $\mathbf{P}_l^h$ the k subjects from the train-in group with the smallest predicted difference under f and increment a frequency map (i.e., a counter) that is initialized at the start of the sample selection process. The frequency value of a subject is then the number of times it was one of the top k predictive samples. These frequencies give an approximated ranking by average distance in target score to all samples.

After the cross-validation is finished, we extract the top k samples[4] with the highest cumulative frequencies. We expect these samples to have the highest representative power as they consistently predicted samples in different holdout groups with low error.

[4] Note that we could pick a different number here. We leave exploring possible other choices for future work.

2.4 Training Process

In the following, we explain how we integrate the sample selection method into the training process of RegGNN. The whole pipeline is shown in Fig. 2.

Given the data set of connectomes our proposed training pipeline consists of the following steps A-C.

A- Training-test split. First, we split the data set into a training and a test set. The test set is used *only* for the final evaluation of RegGNN.

B- Learning-based sample selection. Then, we select the top k samples with the highest representative power from the training set by applying the sample selection method from Sec. 2.3.

C- RegGNN architecture for regression. Finally, we train RegGNN on the top k samples using cross-validation to evaluate model generalizability against perturbations of training and testing data distributions. The final testing is done on the unseen test set.

3 Results and Discussion

We used the pipeline from Sec. 2.4 to predict the full scale intelligence quotient (FIQ) and the verbal intelligence quotient (VIQ) from brain connectomes for both neurotypical (NT) subjects as well as subjects with autism spectrum disorder (ASD). In the following, we summarize these experiments.

Dataset. We used the Autism Brain Imaging Data Exchange (ABIDE) Preprocessed dataset (Craddock et al., 2013) for our experiments. It contains data from 16 imaging sites, preprocessed by five different teams using four pipelines: the Connectome Computation System (CCS), the Configurable Pipeline for the Analysis of Connectomes (CPAC), the Data Processing Assistant for rs-fMRI (DPARSF) and the NeuroImaging Analysis Kit. The preprocessed data sets are available online[5]. To account for possible biases due to differences in sites, we used randomly sampled subsets of the available data for both cohorts; the same sets were also used by (Dryburgh et al., 2020). The NT cohort consisted of 226 subjects (with mean age = (15 ± 3.6)), while the ASD cohort was made up of 202 subjects (with mean age = (15.4 ± 3.8)). FIQ and VIQ scores in the NT cohort have means 111.573 ± 12.056 and 112.787 ± 12.018, whereas FIQ and VIQ scores in the ASD cohort have means 106.102 ± 15.045 and 103.005 ± 16.874, respectively. The brain connectomes were obtained from resting-state functional magnetic resonance imaging using the parcellation from (Tzourio-Mazoyer et al., 2002) with 116 ROIs. The functional connectomes are represented by 116-by-116 matrices, whose entry in row i and column j is the Pearson correlation between the average rs-fMRI signal measured in ROI i and ROI j.

Software. All experiments are done in Python 3.7.10. We used Scikit-learn 0.24.2 (Pedregosa et al., 2011) for machine learning models and PyTorch Geometric 1.6.3 (Fey and Lenssen, 2019) for graph neural network implementations. For Riemannian geometric computations in the SPD space, we used the SPD class from the Morphomatics package of (Ambellan et al., 2021). To extract the graph topological features from the tangent matrices we used NetworkX (Hagberg et al., 2008).

Parameter settings. We trained our method with Adam optimizer (Kingma and Ba, 2017) for 100 epochs with a learning rate of 0.001 and weight decay at 0.0005 based on our empirical observations. The dropout rate after the first graph convolutional layer was set to 0.1. To regularize the adjacency matrices, we used $\mu = 10^{-10}$ in (4). In order to explore the parameter space for the number of selected training samples k, we varied it between 2 and 15.

Evaluation and comparison methods. To test the generalizability and robustness of our method, we used 3-fold cross-validation on both NT and ASD cohorts separately for both FIQ and VIQ prediction. We report the mean absolute error (MAE) and the root mean squared error (RMSE) for all methods. For the sample selection methods, we additionally give the mean, standard deviation, minima and maxima over all tested $k = 2, \ldots, 15$ to test our sample selection methods sensitivity to the selection of k.

To benchmark against our method, we chose state-of-the-art methods from both deep learning and machine learning. The first baseline was CPM (Shen et al., 2017), which was specifically designed for behavioral score prediction on brain connectomes; the second being PNA (Corso et al., 2020), which outperformed common GNNs on both artificial and real-world benchmark regression tasks (but has not been applied to brain connectomes yet). PNA comes with *principal neighborhood aggregation* layers that are defined similarly to graph convolution operations. They are designed to increase the amount of information that is used from the local neighborhoods in the graphs. In our experiments, we inserted PNA layers in our RegGNN architecture. We implemented both a simpler setup with sum aggregation and identity scaling only (denoted by PNA-S), as well as various aggregation (sum, mean, var and max) and scaling (identity, amplification and attenuation) methods (denoted by PNA-V) as detailed in the

[5] http://preprocessed-connectomes-project.org/abide/

paper of (Corso et al., 2020). The code of both CPM[6] and PNA[7] is available online.

In order to assess the effect of the sample selection method, we also always trained each architecture on *all* samples as a baseline.

Evaluation of the sample selection. For each architecture, we compared several methods that can be used as measure of difference in the sample selection (viz., Sec. 2.3 part (*ii*)) to train the linear mapping f.

The first class of methods was the proposed one: we encoded the differences via tangent matrices in the SPD space. To identify a good choice for handling the information that is contained in the tangent matrices, we compared several methods. As one option, we trained f on the vectorized upper triangular part (including the diagonal) of the tangent matrix; this method is denoted by (tm). Since the matrices are symmetric, ignoring the lower part speeds up computations while not losing information. Further, we used degree centrality (1), eigenvector centrality (2), and closeness centrality (3) and applied them to the tangent matrices; they are denoted by (dc), (ec), and (cc), respectively. The mapping f was then trained on the resulting centrality vectors. Additionally, we tested whether the concatenation of the above centrality measures into a single vector is even more informative. To this end, we used both an unscaled and a scaled version, denoted by (cnu) and (cns) respectively. The unscaled version was generated by simple concatenation of the three feature vectors. However, as the three centrality measures have different ranges, we additionally tested scaling each feature vector first. For this, we used min-max scaling. Remember that min-max scaling of a vector v is defined elementwise by

$$\widetilde{v_i} := \frac{v_i - \max(v)}{\max(v) - \min(v)}.$$

Each centrality vector was scaled before concatenating, which then gave a vector with elements in $[0, 1]$ as data for the regression.

We complemented these methods with two baselines. In order to check whether the additional directional information that the tangent matrices contain helps, we also tested whether it suffices to train f on the Riemannian geometric distances $d(\mathbf{P}_i^s, \mathbf{P}_j^s)$ between the connectomes from the train-in group alone; this method is denoted by (g). To assess whether we improve by using the manifold structure of the SPD space at all, we trained f on the Euclidean absolute distance between the upper triangular parts $\widehat{\mathbf{P}}_i^s, \widehat{\mathbf{P}}_j^s$ of each pair of connectomes $\mathbf{P}_i^s, \mathbf{P}_j^s$, i.e., on the scalars $\|\widehat{\mathbf{P}}_i^s - \widehat{\mathbf{P}}_j^s\|_{\mathrm{F}}$ (F standing for the Frobenius norm); we denote this method by (a).

We report the p-value between the best performing sample selection method MAE and the baseline MAE according to a t-test for all architectures.

Results. The results for the NT and ASD cohorts are shown in Tables 2 and 3, respectively. We observe that while the state-of-the-art machine learning model CPM surpasses naive applications of GNNs in the form of PNA, our RegGNN, paired with sample selection, outperforms CPM in all tasks according to both MAE and RMSE with the exception of the NT (FIQ) task. Improvements by our method are especially visible in the ASD cohort. Interestingly, we see that the results of all methods are worse on the ASD cohort compared to the NT cohort. This was also observed by (Dryburgh et al., 2020). We hypothesize that the difficulty of predicting IQ scores in ASD cohort might be caused by the inter-subject heterogeneity that is characteristic for ASD (Tordjman et al., 2018). Another factor may be that ASD samples from ABIDE are biased towards high-functioning individuals (Craddock et al., 2013).

We observe further that sample selection improved the results for RegGNN in all tasks except ASD (VIQ), and for CPM in the ASD cohort. For PNA based architectures, there are drastic improvements in NT (FIQ) and ASD (VIQ) and incremental improvements in the remaining other tasks. An exception to this is the PNA-V setup for ASD (FIQ), where most models with sample selection perform worse than the one trained on all samples. This might be partly explained by the more complicated structure of PNA with various aggregation models, which might demand more samples for correct training. For all models, we see that the minimum MAE over k is lower than the MAE version that was trained on the full data sets even when the mean MAE across k is higher in all tasks. This indicates that improvements are highly likely with fine-tuning of parameter k. Our experiments did not reveal a clear trend for the value of k for which the minimum was attained.

Improvements to the performance of CPM are not statistically significant ($p = 0.87$ for ASD (FIQ), $p = 0.15$ for ASD (VIQ)). Similarly, we observed that improvements to the performance of RegGNN are only statistically significant in NT (VIQ) task ($p < 0.05$ for NT (VIQ), $p = 0.98$ for NT (FIQ), $p = 0.64$ for ASD (FIQ)). The performance increases for PNA models are more consistent, as PNA-S improved significantly in three out of four tasks ($p < 0.01$ for NT (FIQ), $p = 0.11$ for NT (VIQ), $p < 0.05$ for ASD (FIQ), $p < 0.01$ for ASD (VIQ)), and PNA-V improved significantly in two out of four tasks ($p < 0.05$ for NT (FIQ), $p = 0.21$ for

[6] https://github.com/esfinn/cpm_tutorial
[7] https://github.com/lukecavabarrett/pna/

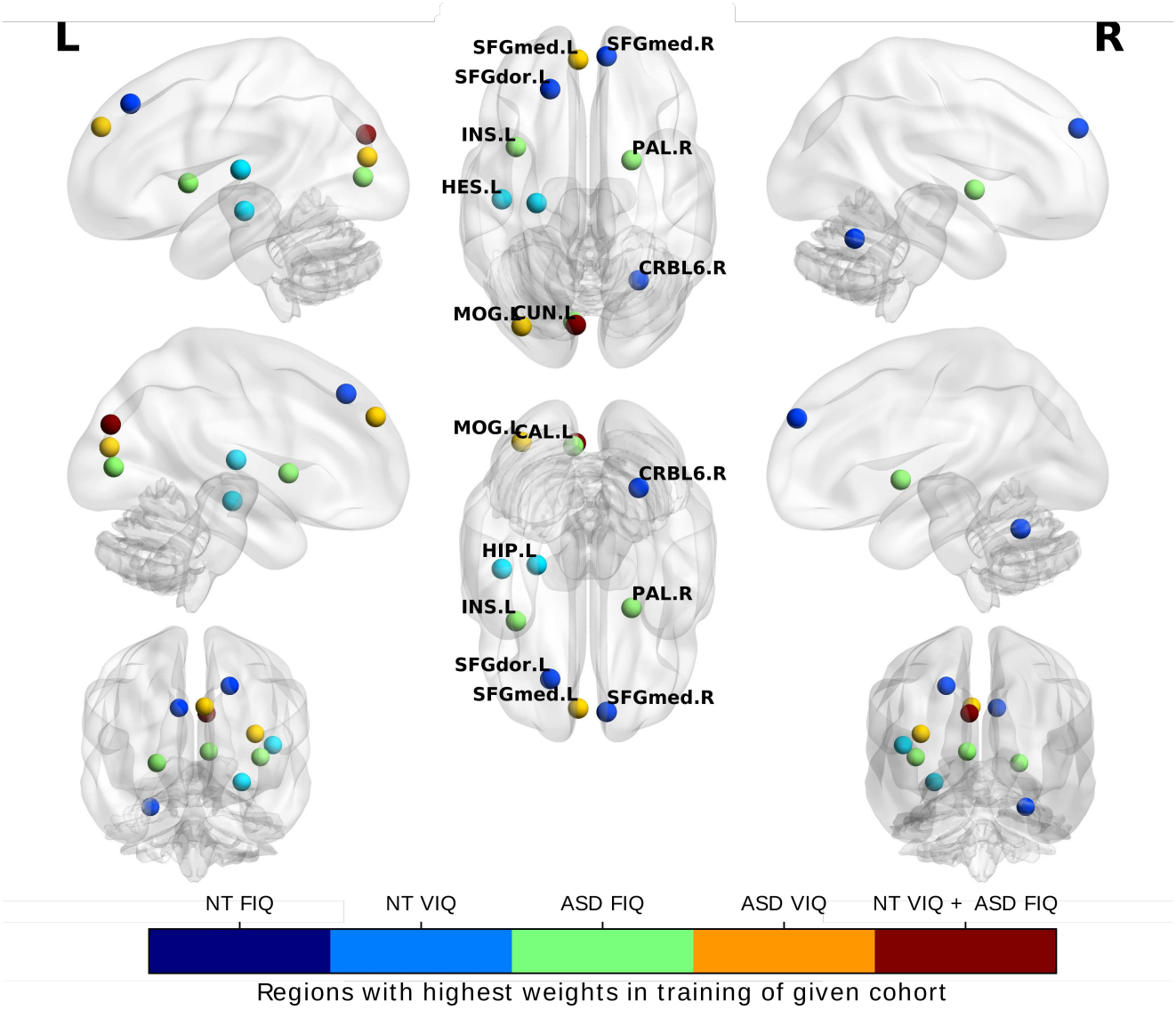

Fig. 3: The top three most relevant brain regions for each task according to the learned weights extracted from the last layer of the trained RegGNN. The colors indicate the task. NT: neurotypical subjects. ASD: autism spectrum disorder subjects. FIQ: fluid intelligence quotient. VIQ: verbal intelligence quotient.

NT (VIQ), $p = 0.11$ for ASD (FIQ), $p < 0.01$ for ASD (VIQ)).

Within the sample selection pipelines, the best performing methods always utilize the Riemannian geometric structure of the SPD space with respect to MAE, apart from PNA-V results for the NT (VIQ) task. In the majority of cases, the methods that rely on tangent matrices perform best with the vectorized version of the whole tangent matrix being the best method in NT (VIQ) and ASD (FIQ). We also see that the three centrality measures and concatenated versions perform well consistently. Our results do not reveal any finer pattern among the sample selection measures, but we can conclude that using Riemannian structure of connectomes to estimate their predictive power outperforms methods that do not leverage these geometric properties. Thus, other metrics should also be considered when deciding for one. The tangent matrix method seems to perform very well but is also the most time consuming since no dimension reduction is performed. On the contrary, computing centrality measures significantly reduces the size of the matrices which speeds up the process sufficiently. In our experiments, we observed that training linear regression models using tangent matrices took up to 16 times more time compared to training models using centrality measures. To understand the latter methods and how they work better, it would be helpful to analyze their behavior on the tan-

gent matrices mathematically. In contrast to their use for adjacency matrices of graphs, this is, to the best of our knowledge, unknown. It is thus an interesting venue for future work.

An important advantage of using sample selection in training graph neural networks is the decrease in the computational power needed for the training process. Using the computational power more efficiently leads to shorter training times on fixed amount of data, which opens up opportunities to train more complex models on more data or in shorter amounts of time. While the exact time required for sample selection is heavily dependent on the hardware used and varies based on the model architecture, number of epochs in training, number of training samples, and the number k of samples to select, our observations during the experiments showed that sample selection reduced the training time by 20% on average. Therefore, usage of our sample selection pipeline can enable the use of deeper neural network architectures on connectomes and provides a topic of interest in future work.

Explainability and biomarker discovery. In order to determine the regions of interest in the brain that influence the prediction most, we extracted for each of the four tasks the learned weights of the RegGNN using the best-performing sample selection method. In Fig. 3, we show the regions of interest with the 3 highest weights averaged over $k = 2, \ldots, 15$; underlying is the AAL parcellation atlas (Tzourio-Mazoyer et al., 2002).[8] For the FIQ prediction task in the NT cohort, we see that the left superior dorsal frontal gyrus (SFGdor.L), right superior frontal medial gyrus (SFGmed.R), and right cerebellum 6 (CRBL6.R) have the highest weights. For the VIQ prediction task in the same cohort, left hippocampus (HIP.L), left heschl gyrus (HES.L), and left cuneus (CUN.L) possess the highest weights. In the ASD cohort, the highest weights for FIQ prediction are left insula (INS.L), left calcarine cortex (CAL.L), and right pallidum (PAL.R), while the highest weights for VIQ prediction have the left superior frontal medial gyrus (SFGmed.L) left middle occipital gyrus (MOG.L), and left cuneus (CUN.L).

According to our results, the more important regions of interest in IQ prediction lie in the left hemisphere of the brain. Our findings are in line with other studies, that found that the insula shows greater activity in various cognitive tasks (Critchley et al., 2000) and that the surface areal change in the left cuneus correlates strongly with full IQ, especially in perceptual tasks in young adults with very low birth weight (Skranes et al., 2013). Furthermore, we observe that left cuneus was influential in predicting VIQ in both cohorts. Finally, as (Dryburgh et al., 2020), our experiments indicate that the middle frontal gyrus is a significant region in IQ prediction.

[8] The brain networks were visualized with the BrainNet Viewer (Xia et al., 2013).

4 Conclusion

In this work, we applied RegGNN, a new graph neural network, to connectome data of neurotypical subjects and subjects with autism spectrum disorder to predict full scale and verbal intelligence quotients. We trained it using a novel sample selection method, which tries to identify samples within the training set that are expected to better predict the cognitive scores of new subjects. This enabled us to train the network with only 15 samples or less, while the testing performance was on par or even better than state-of-the-art methods for cognitive score prediction from connectomes. Both the sample selection and RegGNN are easy to implement in open access software and can be used in clinical practice.

Declarations

Funding This work was funded by generous grants from the European H2020 Marie Sklodowska-Curie action (grant no. 101003403, http://basira-lab.com/normnets/) to I.R. and the Scientific and Technological Research Council of Turkey to I.R. under the TUBITAK 2232 Fellowship for Outstanding Researchers (no. 118C288, http://basira-lab.com/reprime/). However, all scientific contributions made in this project are owned and approved solely by the authors. M.A.G is funded by the same TUBITAK 2232 Fellowship. M.H. is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689).

Conflicts of interest/Competing interests The authors declare that they have no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication All authors agreed to the publication of this article.

Availability of data and material The ABIDE data that was used in this work is available online; the link is given in the text.

Code availability All methods were implemented in Python. Our RegGNN code is available at https://github.com/basiralab/RegGNN.

Author Contributions Author contributions included designing the GNN and sample selection method as well as

the experimental setup (all authors), implementation of the methods and experiments (MAD and MAG), writing the manuscript (MH) and revising it critically for important intellectual content (all authors), and approval of final version to be published and agreement to be accountable for the integrity and accuracy of all aspects of the work (all authors).

References

Abbas, K., Liu, M., Venkatesh, M., Amico, E., Kaplan, A. D., Ventresca, M., Pessoa, L., Harezlak, J., and Goi, J. (2021). Geodesic distance on optimally regularized functional connectomes uncovers individual fingerprints. *Brain Connect.*, 0(0):null.

Ambellan, F., Hanik, M., and von Tycowicz, C. (2021). Morphomatics: Geometric morphometrics in non-Euclidean shape spaces. https://morphomatics.github.io/.

Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. (2006). Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magn. Reson. Med.*, 56(2):411–421.

Batty, G. D., Deary, I. J., and Gottfredson, L. S. (2007). Premorbid (early life) iq and later mortality risk: Systematic review. *Ann. Epidemiol.*, 17(4):278–288.

Bessadok, A., Mahjoub, M. A., and Rekik, I. (2021). Graph neural networks in network neuroscience. *arXiv preprint arXiv:2106.03535*.

Cole, M. W., Ito, T., and Braver, T. S. (2015). Lateral prefrontal cortex contributes to fluid intelligence through multinetwork connectivity. *Brain Connect.*, 5(8):497–504.

Cole, M. W., Yarkoni, T., Repovš, G., Anticevic, A., and Braver, T. S. (2012). Global connectivity of prefrontal cortex predicts cognitive control and intelligence. *J. Neurosci.*, 32(26):8988–8999.

Colom, R., Escorial, S., Shih, P. C., and Privado, J. (2007). Fluid intelligence, memory span, and temperament difficulties predict academic performance of young adolescents. *Pers. Individ. Differ.*, 42(8):1503 – 1514.

Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Veličković, P. (2020). Principal neighbourhood aggregation for graph nets. *arXiv preprint arXiv:2004.05718*.

Craddock, C., Benhajali, Y., Chu, C., Chouinard, F., Evans, A., Jakab, A., Khundrakpam, B. S., Lewis, J. D., Li, Q., Milham, M., et al. (2013). The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Front. Neuroinform.*, 7.

Critchley, H. D., Daly, E. M., Bullmore, E. T., Williams, S. C. R., Van Amelsvoort, T., Robertson, D. M., Rowe, A., Phillips, M., McAlonan, G., Howlin, P., and Murphy, D. G. M. (2000). The functional neuroanatomy of social behaviour: Changes in cerebral blood flow when people with autistic disorder process facial expressions. *Brain*, 123(11):2203–2212.

Dadi, K., Rahim, M., Abraham, A., Chyzhyk, D., Milham, M., Thirion, B., Varoquaux, G., Initiative, A. D. N., et al. (2019). Benchmarking functional connectome-based predictive models for resting-state fmri. *NeuroImage*, 192:115–134.

Deary, I. J., Strand, S., Smith, P., and Fernandes, C. (2007). Intelligence and educational achievement. *Intelligence*, 35(1):13–21.

Dehmamy, N., Barabasi, A.-L., and Yu, R. (2019). Understanding the representation power of graph neural networks in learning graph topology. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

do Carmo, M. P. (1992). *Riemannian Geometry; 2nd ed.* Birkhuser, Boston, MA.

Dodero, L., Minh, H. Q., Biagio, M. S., Murino, V., and Sona, D. (2015). Kernel-based classification for brain connectivity graphs on the Riemannian manifold of positive definite matrices. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 42–45.

Dryburgh, E., McKenna, S., and Rekik, I. (2020). Predicting full-scale and verbal intelligence scores from functional connectomic data in individuals with autism spectrum disorder. *Brain. Imaging. Behav.*, 14:17691778.

Dubois, J., Galdi, P., Paul, L. K., and Adolphs, R. (2018). A distributed brain network predicts general intelligence from resting-state human neuroimaging data. *Philos. Trans. R. Soc. B*, 373(1756):20170284.

Errica, F., Podda, M., Bacciu, D., and Micheli, A. (2019). A fair comparison of graph neural networks for graph classification. *arXiv preprint arXiv:1912.09893*.

Faraut, J. and Korányii, A. (1994). *Analysis on symmetric cones*. Oxford University Press, New York, USA.

Fey, M. and Lenssen, J. E. (2019). Fast graph representation learning with pytorch geometric.

Fornito, A., Zalesky, A., and Bullmore, E. (2016). *Fundamentals of brain network analysis*. Academic Press.

Garcia, V. and Bruna, J. (2017). Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*.

Gottfredson, L. S. and Deary, I. J. (2004). Intelligence predicts health and longevity, but why? *Curr. Dir. Psychol. Sci.*, 13(1):1–4.

Gray, J. R., Chabris, C. F., and Braver, T. S. (2003). Neural mechanisms of general fluid intelligence. *Nat. Neurosci.*, 6(3):316–322.

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA.

He, T., Kong, R., Holmes, A. J., Nguyen, M., Sabuncu, M. R., Eickhoff, S. B., Bzdok, D., Feng, J., and Yeo, B. T. (2020). Deep neural networks and kernel regression achieve comparable accuracies for functional connectivity prediction of behavior and demographics. *NeuroImage*, 206:116276.

Jiang, R., Calhoun, V. D., Fan, L., Zuo, N., Jung, R., Qi, S., Lin, D., Li, J., Zhuo, C., Song, M., et al. (2020). Gender differences in connectome-based predictions of individualized intelligence quotient and sub-domain scores. *Cereb. Cortex*, 30(3):888–900.

Kim, J., Kim, T., Kim, S., and Yoo, C. D. (2019). Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11–20.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks.

Mcdaniel, M. (2005). Big-brained people are smarter: A meta-analysis of the relationship between in vivo brain volume and intelligence. *Intelligence*, 33:337–346.

Moakher, M. (2005). A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.*, 26(3):735747.

Pamplona, G. S. P., Santos Neto, G. S., Rosset, S. R. E., Rogers, B. P., and Salmon, C. E. G. (2015). Analyzing the association between functional connectivity of the brain and intellectual performance. *Front. Hum. Neusci.*, 9:61.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830.

Pennec, X., Fillard, P., and Ayache, N. (2006). A Riemannian framework for tensor computing. *Int. J. Comput. Vis.*, 66(1):41–66.

Qi, X., Liao, R., Jia, J., Fidler, S., and Urtasun, R. (2017). 3d graph neural networks for rgbd semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5199–5208.

Qu, M., Bengio, Y., and Tang, J. (2019). Gmnn: Graph markov neural networks. In *International conference on machine learning*, pages 5241–5250. PMLR.

Reiss, A. L., Abrams, M. T., Singer, H. S., Ross, J. L., and Denckla, M. B. (1996). Brain development, gender and iq in children: a volumetric imaging study. *Brain*, 119(5):1763–1774.

Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A. (2020). Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Shen, X., Finn, E., Scheinost, D., Rosenberg, M., Chun, M., Papademetris, X., and Constable, R. (2017). Using connectome-based predictive modeling to predict individual behavior from brain connectivity. *Nat. Protoc.*, 12(3):506–518.

Skranes, J., Lhaugen, G. C., Martinussen, M., Hberg, A., Brubakk, A.-M., and Dale, A. M. (2013). Cortical surface area and iq in very-low-birth-weight (vlbw) young adults. *Cortex*, 49(8):2264–2271.

Sporns, O., Tononi, G., and Ktter, R. (2005). The human connectome: A structural description of the human brain. *PLoS Comput. Biol.*, 1(4).

Tordjman, S., Cohen, D., Anderson, G., Botbol, M., Canitano, R., Coulon, N., and Roubertoux, P. (2018). Repint of reframing autism as a behavioral syndrome and not a specific mental disorder: Implications of genetic and phenotypic heterogeneity. *Neurosci. Biobehav. Rev.*, 89:132–150.

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., Mazoyer, B., and Joliot, M. (2002). Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *NeuroImage*, 15(1):273–289.

Wong, E., Anderson, J. S., Zielinski, B. A., and Fletcher, P. T. (2018). Riemannian regression and classification models of brain networks applied to autism. In *Connectomics in NeuroImaging*, pages 78–87, Cham. Springer International Publishing.

Woolgar, A., Parr, A., Cusack, R., Thompson, R., Nimmo-Smith, I., Torralva, T., Roca, M., Antoun, N., Manes, F., and Duncan, J. (2010). Fluid intelligence loss linked to restricted regions of damage within frontal and parietal cortex. *Proc. Natl. Acad.*

Sci. U.S.A., 107(33):14899–14902.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE T. Neur. Net. Lear.*, 32:4–24.

Xia, M., Wang, J., and He, Y. (2013). Brainnet viewer: a network visualization tool for human brain connectomics. *PloS one*, 8(7):e68910.

Yamin, M. A., Tessadori, J., Akbar, M. U., Dayan, M., Murino, V., and Sona, D. (2020). Geodesic clustering of positive definite matrices for classification of mental disorder using brain functional connectivity. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–5.

You, K. and Park, H.-J. (2021). Re-visiting Riemannian geometry of symmetric positive definite matrices for the analysis of functional connectivity. *NeuroImage*, 225:117464.

Yun, S., Jeong, M., Kim, R., Kang, J., and Kim, H. J. (2019). Graph transformer networks. *arXiv preprint arXiv:1911.06455*.

Zhang, C., Song, D., Huang, C., Swami, A., and Chawla, N. V. (2019). Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 793–803.

Method	Mean MAE ± std (min, max)	Mean RMSE ± std (min, max)
NT (FIQ)		
CPM	**9.672**	**12.440**
CPM (a)	11.383 ± 4.806 (9.589, 28.675)	13.972 ± 4.986 (12.075, 31.911)
CPM (g)	10.546 ± 1.407 (9.729, 14.989)	13.358 ± 1.831 (12.393, 19.358)
CPM (tm)	10.245 ± 2.086 (9.523, 17.745)	12.959 ± 2.515 (12.096, 22.009)
CPM (dc)	10.577 ± 2.487 (9.561, 19.394)	13.525 ± 2.751 (12.131, 23.210)
CPM (ec)	10.537 ± 2.131 (9.595, 18.159)	13.456 ± 2.393 (12.366, 21.976)
CPM (cc)	10.074 ± 1.307 (**9.370**, 14.741)	12.931 ± 1.669 (**12.065**, 18.895)
CPM (cnu)	10.895 ± 4.402 (9.463, 26.749)	13.775 ± 4.633 (12.194, 30.465)
CPM (cns)	11.262 ± 4.585 (9.623, 27.771)	14.156 ± 4.822 (12.406, 31.521)
PNA-S	17.217	21.008
PNA-S (a)	12.267 ± 0.948 (11.002, 14.480)	15.338 ± 1.079 (13.914, 18.110)
PNA-S (g)	12.305 ± 1.246 (10.814, 15.267)	15.474 ± 1.357 (13.897, 18.777)
PNA-S (tm)	**11.537 ± 0.727** (**10.639**, 12.890)	**14.466 ± 0.814** (**13.322**, 15.853)
PNA-S (dc)	12.211 ± 1.119 (10.751, 14.012)	15.389 ± 1.289 (13.619, 17.191)
PNA-S (ec)	12.124 ± 1.307 (10.956, 16.118)	15.270 ± 1.466 (13.865, 19.509)
PNA-S (cc)	12.166 ± 0.898 (10.727, 13.895)	15.413 ± 1.098 (13.615, 17.309)
PNA-S (cnu)	12.608 ± 1.204 (10.935, 15.944)	15.799 ± 1.378 (13.827, 19.684)
PNA-S (cns)	12.509 ± 0.963 (11.152, 14.747)	15.737 ± 1.172 (13.960, 18.528)
PNA-V	20.109	25.113
PNA-V (a)	15.979 ± 3.730 (11.607, 26.483)	19.617 ± 4.111 (14.575, 30.983)
PNA-V (g)	14.570 ± 4.115 (**11.014**, 26.962)	18.020 ± 4.329 (**13.984**, 30.811)
PNA-V (tm)	15.450 ± 4.313 (11.259, 26.906)	19.030 ± 4.804 (14.273, 32.059)
PNA-V (dc)	**14.300 ± 2.111** (11.475, 17.750)	**17.900 ± 2.588** (14.313, 22.516)
PNA-V (ec)	15.385 ± 3.141 (11.504, 21.710)	19.294 ± 3.905 (14.779, 27.041)
PNA-V (cc)	15.288 ± 3.023 (11.709, 20.592)	18.845 ± 3.378 (14.770, 24.803)
PNA-V (cnu)	16.800 ± 5.053 (12.352, 28.022)	20.470 ± 5.736 (15.180, 32.794)
PNA-V (cns)	15.910 ± 5.976 (12.507, 36.704)	19.671 ± 6.875 (15.478, 43.397)
RegGNN	9.768	**12.270**
RegGNN (a)	10.360 ± 1.090 (9.624, 14.027)	12.997 ± 1.278 (12.158, 17.335)
RegGNN (g)	10.032 ± 0.330 (9.576, 10.790)	12.563 ± 0.310 (12.148, 13.311)
RegGNN (tm)	9.820 ± 0.512 (9.525, 11.613)	12.378 ± 0.528 (12.116, 14.259)
RegGNN (dc)	**9.714 ± 0.332** (9.485, 10.634)	12.531 ± 0.458 (**12.064**, 13.716)
RegGNN (ec)	9.815 ± 0.245 (9.469, 10.334)	12.609 ± 0.334 (12.171, 13.286)
RegGNN (cc)	9.777 ± 0.391 (9.461, 10.757)	12.516 ± 0.425 (12.121, 13.523)
RegGNN (cnu)	9.745 ± 0.451 (**9.438**, 11.018)	12.482 ± 0.473 (12.085, 13.758)
RegGNN (cns)	9.716 ± 0.292 (9.452, 10.474)	12.466 ± 0.217 (12.207, 12.982)
NT (VIQ)		
CPM	**9.517**	**12.049**
CPM (a)	11.035 ± 4.970 (9.481, 28.895)	13.728 ± 5.243 (12.031, 32.518)
CPM (g)	10.363 ± 1.682 (9.549, 16.146)	13.253 ± 2.126 (12.236, 20.633)
CPM (tm)	10.194 ± 2.316 (9.503, 18.543)	12.922 ± 2.928 (12.035, 23.477)
CPM (dc)	10.065 ± 1.760 (9.391, 16.386)	12.767 ± 2.386 (11.877, 21.332)
CPM (ec)	9.942 ± 1.768 (**9.323**, 16.275)	12.614 ± 2.209 (**11.828**, 20.532)
CPM (cc)	10.233 ± 1.264 (9.514, 14.667)	12.911 ± 1.595 (12.035, 18.505)
CPM (cnu)	10.544 ± 3.585 (9.480, 23.469)	13.256 ± 4.058 (12.022, 27.885)
CPM (cns)	10.583 ± 3.560 (9.526, 23.416)	13.326 ± 3.891 (12.108, 27.343)
PNA-S	12.838	16.130
PNA-S (a)	11.846 ± 0.942 (10.795, 13.764)	15.057 ± 1.099 (13.824, 17.279)
PNA-S (g)	12.439 ± 1.183 (10.913, 14.774)	15.648 ± 1.322 (13.895, 18.469)
PNA-S (tm)	12.489 ± 3.099 (10.759, 23.367)	15.884 ± 4.323 (13.983, 31.268)
PNA-S (dc)	**11.694 ± 0.814** (**10.302**, 12.901)	**14.707 ± 0.904** (**13.126**, 16.115)
PNA-S (ec)	12.091 ± 1.006 (10.543, 14.032)	15.122 ± 1.061 (13.436, 17.200)
PNA-S (cc)	13.074 ± 1.693 (11.258, 17.486)	16.326 ± 1.795 (14.474, 20.701)
PNA-S (cnu)	12.682 ± 1.308 (10.857, 15.196)	15.950 ± 1.536 (13.795, 18.775)
PNA-S (cns)	12.014 ± 0.859 (10.545, 14.031)	15.141 ± 0.944 (13.503, 17.302)
PNA-V	14.695	18.903
PNA-V (a)	**14.107 ± 2.081** (11.923, 19.482)	**17.696 ± 2.693** (14.832, 25.550)
PNA-V (g)	14.983 ± 5.211 (11.639, 32.479)	18.681 ± 6.224 (14.715, 39.771)
PNA-V (tm)	15.489 ± 4.211 (11.717, 27.980)	19.157 ± 4.775 (14.624, 33.183)
PNA-V (dc)	14.332 ± 2.931 (**11.188**, 20.545)	18.170 ± 3.917 (**14.284**, 26.511)
PNA-V (ec)	14.924 ± 3.424 (11.360, 21.703)	18.539 ± 4.185 (14.392, 27.289)
PNA-V (cc)	16.049 ± 4.185 (11.237, 24.616)	20.035 ± 4.966 (14.408, 29.373)
PNA-V (cnu)	14.805 ± 2.587 (11.890, 21.107)	18.344 ± 2.883 (15.171, 25.473)
PNA-V (cns)	14.915 ± 2.845 (11.898, 22.045)	18.674 ± 3.612 (14.895, 28.544)
RegGNN	10.195	13.044
RegGNN (a)	9.587 ± 0.206 (9.477, 10.311)	12.223 ± 0.299 (12.054, 13.261)
RegGNN (g)	9.779 ± 0.126 (9.551, 9.964)	12.530 ± 0.199 (12.190, 12.803)
RegGNN (tm)	**9.514 ± 0.036** (9.471, 9.594)	**12.041 ± 0.035** (**12.004**, 12.140)
RegGNN (dc)	9.639 ± 0.161 (9.468, 10.000)	12.213 ± 0.235 (12.008, 12.707)
RegGNN (ec)	9.599 ± 0.243 (**9.453**, 10.189)	12.199 ± 0.274 (12.020, 12.846)
RegGNN (cc)	9.711 ± 0.193 (9.499, 10.161)	12.313 ± 0.284 (12.023, 13.062)
RegGNN (cnu)	9.521 ± 0.042 (9.473, 9.651)	12.134 ± 0.050 (12.050, 12.266)
RegGNN (cns)	9.581 ± 0.153 (9.497, 10.109)	12.236 ± 0.199 (12.102, 12.907)

Table 2: Comparison of regression methods on the NT cohort. The best performing method for each architecture is bold while the second best is underlined. The mean ± standard deviation as well as minima and maxima over $k = 2, \ldots, 15$ (in brackets) are given. The overall best performing method according to mean error and the best sample selection performance are indicated in blue.

Method	Mean MAE ± std (min, max)	Mean RMSE ± std (min, max)
ASD (FIQ)		
CPM	12.533	15.965
CPM (a)	13.673 ± 3.748 (12.082, 26.912)	16.783 ± 4.048 (15.043, 31.074)
CPM (g)	13.267 ± 1.367 (12.166, 17.792)	16.728 ± 1.672 (15.433, 22.297)
CPM (tm)	**12.464 ± 1.090** (12.093, 16.386)	**15.522 ± 1.336** (15.084, 20.333)
CPM (dc)	12.644 ± 1.498 (12.082, 18.027)	15.685 ± 1.723 (15.116, 21.887)
CPM (ec)	12.825 ± 1.400 (**12.074**, 17.527)	15.922 ± 1.654 (**15.017**, 21.252)
CPM (cc)	13.310 ± 1.331 (12.430, 17.821)	16.703 ± 1.654 (15.618, 22.335)
CPM (cnu)	12.934 ± 2.460 (12.094, 21.790)	16.039 ± 2.817 (15.086, 26.182)
CPM (cns)	12.742 ± 1.527 (12.166, 18.129)	15.796 ± 1.838 (15.077, 22.298)
PNA-S	17.162	22.023
PNA-S (a)	15.260 ± 0.865 (13.947, 17.049)	18.768 ± 0.940 (17.527, 20.665)
PNA-S (g)	15.025 ± 1.321 (13.207, 17.762)	18.844 ± 1.641 (16.642, 22.009)
PNA-S (tm)	14.781 ± 1.182 (**13.168**, 18.488)	18.630 ± 1.496 (16.522, 23.426)
PNA-S (dc)	14.248 ± 0.533 (13.595, 15.651)	17.715 ± 0.723 (16.654, 19.644)
PNA-S (ec)	14.758 ± 1.277 (13.207, 18.534)	18.263 ± 1.608 (**16.440**, 23.134)
PNA-S (cc)	14.558 ± 0.666 (13.824, 16.586)	18.126 ± 0.894 (17.296, 20.806)
PNA-S (cnu)	14.038 ± 0.434 (13.353, 14.915)	**17.438 ± 0.580** (16.469, 18.642)
PNA-S (cns)	**14.023 ± 0.401** (13.465, 14.782)	17.620 ± 0.550 (16.853, 18.605)
PNA-V	15.671	20.085
PNA-V (a)	17.386 ± 2.428 (14.453, 23.263)	21.234 ± 2.922 (18.006, 28.505)
PNA-V (g)	17.611 ± 2.228 (14.770, 23.572)	22.035 ± 3.023 (18.422, 30.646)
PNA-V (tm)	18.458 ± 5.559 (**13.770**, 32.776)	23.112 ± 6.498 (17.326, 38.914)
PNA-V (dc)	16.358 ± 2.793 (14.045, 24.543)	20.325 ± 3.344 (17.355, 29.689)
PNA-V (ec)	17.085 ± 4.124 (13.778, 28.115)	21.095 ± 4.885 (**17.101**, 33.584)
PNA-V (cc)	16.172 ± 1.181 (14.541, 18.242)	19.942 ± 1.346 (17.970, 22.102)
PNA-V (cnu)	19.552 ± 6.530 (13.840, 33.875)	24.470 ± 8.514 (17.399, 44.199)
PNA-V (cns)	**15.601 ± 2.041** (13.820, 20.748)	**19.406 ± 2.359** (17.484, 25.427)
RegGNN	12.564	15.624
RegGNN (a)	12.588 ± 0.541 (12.170, 13.825)	15.628 ± 0.598 (15.102, 17.006)
RegGNN (g)	12.977 ± 1.073 (12.137, 16.292)	16.194 ± 1.358 (15.146, 20.300)
RegGNN (tm)	**12.148 ± 0.072** (12.078, 12.379)	**15.130 ± 0.067** (15.074, 15.355)
RegGNN (dc)	12.247 ± 0.167 (12.073, 12.667)	15.214 ± 0.178 (15.058, 15.693)
RegGNN (ec)	12.361 ± 0.246 (12.106, 12.905)	15.320 ± 0.265 (**15.040**, 15.895)
RegGNN (cc)	12.720 ± 0.466 (12.134, 14.008)	15.756 ± 0.560 (15.074, 17.321)
RegGNN (cnu)	12.301 ± 0.226 (**12.064**, 12.804)	15.331 ± 0.250 (15.120, 15.881)
RegGNN (cns)	12.300 ± 0.159 (12.132, 12.763)	15.306 ± 0.201 (15.114, 15.894)
ASD (VIQ)		
CPM	14.171	18.834
CPM (a)	15.006 ± 5.390 (12.920, 34.315)	19.034 ± 5.458 (16.887, 38.586)
CPM (g)	14.336 ± 3.490 (**12.669**, 26.704)	18.646 ± 3.660 (16.845, 31.560)
CPM (tm)	15.496 ± 2.494 (13.694, 21.575)	20.191 ± 3.140 (17.994, 28.150)
CPM (dc)	14.449 ± 1.882 (13.289, 19.795)	18.767 ± 2.166 (17.421, 24.648)
CPM (ec)	**13.697 ± 1.389** (12.824, 18.258)	**17.985 ± 1.737** (**16.838**, 23.693)
CPM (cc)	14.954 ± 3.088 (13.432, 25.694)	19.613 ± 3.506 (17.508, 31.570)
CPM (cnu)	14.698 ± 1.565 (13.046, 19.980)	18.917 ± 1.715 (17.412, 24.674)
CPM (cns)	15.417 ± 2.761 (13.593, 22.869)	20.041 ± 4.128 (17.648, 32.850)
PNA-S	19.955	25.848
PNA-S (a)	15.993 ± 1.096 (**14.125**, 18.861)	20.500 ± 1.284 (18.349, 23.726)
PNA-S (g)	**15.540 ± 1.176** (14.139, 18.017)	**20.037 ± 1.548** (**18.256**, 23.559)
PNA-S (tm)	16.766 ± 1.330 (15.020, 20.365)	21.548 ± 1.583 (19.513, 25.735)
PNA-S (dc)	16.820 ± 2.161 (14.332, 21.360)	21.386 ± 2.303 (18.685, 25.724)
PNA-S (ec)	16.445 ± 1.598 (14.518, 21.308)	21.150 ± 1.900 (18.760, 26.560)
PNA-S (cc)	16.113 ± 1.458 (14.600, 19.985)	21.012 ± 1.715 (19.057, 25.231)
PNA-S (cnu)	16.359 ± 1.450 (14.661, 20.737)	20.880 ± 1.714 (18.905, 26.173)
PNA-S (cns)	15.887 ± 0.773 (14.621, 17.091)	20.282 ± 0.843 (18.785, 21.847)
PNA-V	22.518	28.804
PNA-V (a)	18.145 ± 3.017 (14.837, 26.420)	22.937 ± 3.373 (19.233, 31.651)
PNA-V (g)	**16.906 ± 1.566** (14.975, 21.123)	**21.791 ± 2.210** (19.658, 28.209)
PNA-V (tm)	19.040 ± 3.597 (15.631, 30.284)	24.162 ± 4.053 (19.972, 36.701)
PNA-V (dc)	19.734 ± 4.936 (15.055, 33.387)	25.088 ± 6.380 (19.898, 43.769)
PNA-V (ec)	17.837 ± 2.000 (14.973, 21.401)	22.721 ± 2.341 (19.373, 27.477)
PNA-V (cc)	18.663 ± 4.268 (14.969, 32.436)	24.016 ± 5.063 (19.679, 40.173)
PNA-V (cnu)	17.314 ± 2.329 (14.660, 22.610)	22.128 ± 2.772 (19.026, 28.786)
PNA-V (cns)	17.714 ± 2.833 (14.697, 24.781)	22.630 ± 3.748 (**18.958**, 32.707)
RegGNN	**13.090**	**17.250**
RegGNN (a)	13.356 ± 0.443 (12.834, 14.402)	17.272 ± 0.480 (**16.782**, 18.502)
RegGNN (g)	13.316 ± 0.545 (**12.687**, 14.700)	17.474 ± 0.663 (16.877, 19.186)
RegGNN (tm)	14.110 ± 1.237 (13.194, 17.014)	18.288 ± 1.656 (17.130, 22.128)
RegGNN (dc)	14.516 ± 2.043 (13.369, 19.931)	18.381 ± 2.261 (17.140, 24.445)
RegGNN (ec)	13.419 ± 0.705 (12.766, 15.061)	17.526 ± 0.945 (16.815, 19.958)
RegGNN (cc)	14.021 ± 1.001 (13.149, 16.812)	18.731 ± 1.242 (17.563, 21.916)
RegGNN (cnu)	14.001 ± 0.605 (12.762, 14.885)	18.084 ± 0.569 (17.022, 18.924)
RegGNN (cns)	14.020 ± 0.440 (12.950, 14.650)	18.007 ± 0.434 (17.008, 18.571)

Table 3: Comparison of regression methods on the ASD cohort. The best performing method for each architecture is bold while the second best is underlined. The mean ± standard deviation as well as minima and maxima over $k = 2, \ldots, 15$ (in brackets) are given. The overall best performing method according to mean error and the best sample selection performance are indicated in blue.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- BIBChecklist.pdf