PRE-TRAINING GRAPH ATTENTION CONVOLUTION FOR BRAIN STRUCTURAL IMAGING BIOMARKER ANALYSIS AND ITS APPLICATION TO ALZHEIMER'S DISEASE PATHOLOGY IDENTIFICATION

Zhangsihao Yang[1], *Yi Su*[2], *Mohammad Farazi*[1], *Wenhui Zhu*[1], *Yanxi Chen*[1], *Eric M Reiman*[2],
Richard J Caselli[3], Kewei Chen[2], Yalin Wang[1], Natasha Lepore[4]

[1] School of Computing and Augmented Intelligence, Arizona State University, AZ, USA
[2] Banner Alzheimer's Institute, Phoenix, AZ, USA
[3] Department of Neurology, Mayo Clinic Arizona, Scottsdale, AZ, USA
[4] cCIBORG Lab, Department of Radiology Children's Hospital Los Angeles, Los Angeles, CA, USA

ABSTRACT

Biomarkers are one of the primary diagnostic tools to facilitate the early detection of Alzheimer's disease. The accumulation of beta-amyloid (Aβ) plaques in the human brain is one of the presymptomatic hallmarks of AD. However, current methods to detect Aβ pathology are either invasive (lumbar puncture), quite costly, and not widely available (amyloid positron emission tomography - PET) or largely under development (blood-based biomarkers - BBBM). Thus a less invasive and cost-effective approach is demanded. Magnetic resonance imaging (MRI) which has been used widely in preclinical AD, has recently shown the capability to predict brain Aβ positivity. This motivates us to develop a method, pre-training graph attention convolution, taking MRI to predict Aβ positivity. The proposed self-supervised learning architecture refines feature extraction from mesh representation via pre-training and fine-tuning, resulting in more powerful biomarkers for Aβ identification. We obtain subjects from the Alzheimer's Disease Neuroimaging Initiative (ADNI) and use our method to discriminate Aβ positivity. Theoretically, we provide analysis toward the understanding of what the network has learned. Empirically, it shows strong performance on par or even better than state of the art.

1. INTRODUCTION

By 2050, 1 out of 85 people worldwide will be affected by Alzheimer's disease (AD) [1]. Degenerative diseases have put a great burden on worldwide healthcare systems regarding cost and therapies [2]. The accumulation of beta-amyloid (Aβ) plaques in the human brain is one of the presymptomatic hallmarks of AD [3] and was found promising for early detection of AD. Current amyloid measures, such as lumbar puncture and amyloid positron emission tomography (PET), are either invasive or too expensive. Even though Blood-based biomarkers (BBBM) are developed for screening preclinical AD, only weak, or no correlation is found with the estimate of brain amyloid positivity, and [4] BBBM' usefulness for differential diagnosis and prognosis is warranted further development [5]. On the other hand, structural magnetic resonance imaging (sMRI) is widely available in clinical practice. Therefore, there is a strong interest in developing new techniques which predict amyloid burden with sMRI analysis [6].

Recently, the prediction of brain amyloid with sMRI received growing interest because of its wide availability and non-invasive nature. Traditionally, brain image biomarkers [7] use cortical or sub-cortical structure volume. Recent works demonstrate that surface-based brain imaging biomarkers outperform volume measures because they overcome partial volume effects. An example is the radial distance is used as valid hippocampal biomarkers [8].

Meanwhile, because of the high cost and high variability associated with manual data labeling, there are more and more unlabeled 3D data. Traditional studies do not consider unlabeled data, which induces a huge sacrifice of untapped information. Therefore, unsupervised learning attracts more attention and has become an important concept for extracting information from unlabeled data. When we review the trend and development of artificial intelligence, self-supervised training on large datasets and producing pre-trained models for downstream tasks is becoming a predominant power in processing and extracting important features from billions and millions of data [9]. Training an autoencoder on the input data during training has been proven to be an effective method for language processing [10]. In this paper, benefiting from using the mesh data representation, we propose to apply graph attention convolution and point cloud reconstruction to support our self-supervised learning architecture and advance 3D deep learning research.

In our paper, we present a mesh-based framework that is pre-trained on self-analyzing the mesh data, and apply the pre-trained model to large-scale 3D imaging datasets. Our network is designed to be suitable for different kinds of mesh representations to increase flexibility and support a variety of available data. This motivates us to introduce this work designed to pre-train on a large dataset and fine-tune on specific tasks for brain amyloid burden identification. The key contributions of our work are as follows: (1) To our knowledge, this is the first paper using a deep neural network to pre-train and fine-tune for meshes for discoverying brain image biomarkers. (2) With our pre-trained encoder, our self-supervised learning model incorporates unlabeled data into the training stage and enhances the 3D data learning power.

2. METHOD

We take the hippocampus as a composition of two 2D manifolds (Fig. 4). Our work is to extract features on the hippocampal surfaces with graph attention convolution and pre-traing. We first propose our graph attention convolution and the network architecture that could learn features on a graph. Then, we introduce the pipeline of pre-training and fine-tuning. Finally, we briefly explain four baselines used to classify the hippocampus, a commonly-used method that leverages the hippocampal volumes, two state-of-the-art 3D object recognition deep learning approaches, and a support vector machine (SVM) classifier applied to our pre-trained features.

2.1. Graph Attention Convolution

2.1.1. Graph Convolution

Graph Convolution is first introduced in [11]. A graph $\mathcal{G}$ is composed by its adjacency matrix A and features X associated with each node on the graph.

$$H^{(l+1)} = \sigma(\tilde{D}^{(-\frac{1}{2})}\tilde{A}\tilde{D}^{(-\frac{1}{2})}H^{(l)}W^{(l)}) \tag{1}$$

In Equation1, $\tilde{A} = A + I_N$ is the adjacency matrix of the graph $\mathcal{G}$ with added self-connections. I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$ denotes an activation function, such as $ReLU(\cdot) = max(0, \cdot)$. $H^{(l)} \in \mathcal{R}^{N \times D}$ is the matrix of activations in the l^{th} layer; $H^{(0)} = X$. The convolution under the above definition treats the neighboring nodes the same during computation. Dedicated graph convolution is desired for graphs defined by meshes (Section 2.1.2).

2.1.2. Graph Attention Convolution

Graph attention convolution is the core component of our network, as shown in Fig. 1. The layer takes a graph and the features on each node of the graph as input. For each node in the graph, the layer first gathers its neighbors according to an adjacency matrix which could be an n-ring neighbor adjacency matrix in our architecture. We denote r as the feature of the root node and n as the gathered features of the root node and its neighbors. Three linear layers f_V, f_Q, and f_K take n, r, and n as input to compute V, Q, and K. In our work, we keep the output dimension of the key, value, and query fixed at 64.

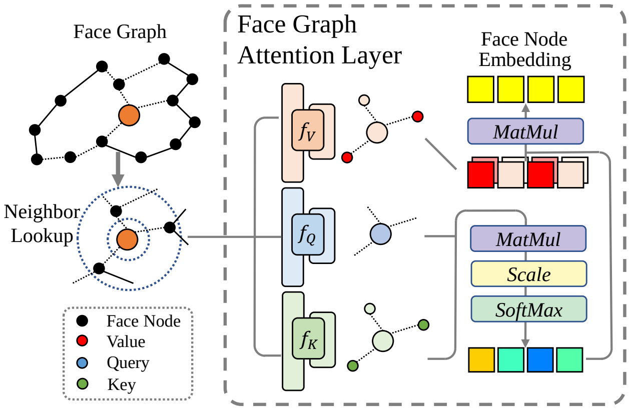

Fig. 1. **Structure of Graph Attention Convolution.** The structure interprets the mesh as a graph, and each node of the graph is a face on the mesh.

$$FaceNodeEmbedding = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

After obtaining V, Q, and K, we use Equation 2 to get the embedding of each face node. In Equation 2, d_k stands for the dimensional size of K. Composing the layers into an encoder is shown in Fig. 3. There are three blocks, and there are 3, 4, and 4 residual blocks in each block. In each residual block, there are two graph attention convolution layers. After the Max Pooling layer, the feature of the mesh is obtained. The overall architecture of our pre-training model is shown in Fig. 3. It has a heavier encoder than the decoder since, after pre-training, we no longer need the decoder. We do not use batch normalization in the decoder. Each graph attention convolution layer has batch normalization and a ReLU layer following.

2.2. Pre-Training and Fine-Tuning

The workflow of our method is shown in Fig. 2. Our method contains two stages, pre-training, and fine-tuning stages.

At the pre-training stage, we define the pre-training task to reconstruct the input mesh. In order to train a model that could reconstruct the input mesh, we need to use the reconstruction loss function. In the reconstruction loss function, we create a reconstruction decoder for this function. The input to this decoder is the graph embedding of the mesh. The expected output is the point cloud sampled from the mesh. We use a network architecture shown in Fig. 3 for decoding a point cloud. So we choose the point cloud as the target for the decoder to generate. And the loss function is the Chamfer Distance (CD), as shown in Equation 3.

$$\mathcal{L}_{CD} = \frac{1}{N}\sum_{n=1}^{N}\min_{\hat{p}\in\hat{s}}\|p_n - \hat{p}\|_2^2 + \frac{1}{M}\sum_{m=1}^{M}\min_{p\in s}\|\hat{p}_m - p\|_2^2 \tag{3}$$

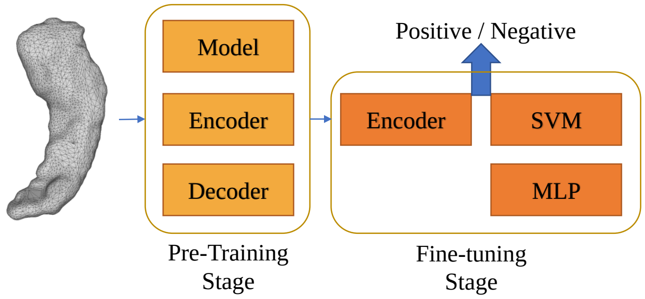

Fig. 2. **The visual comparisons among different data formats to represent a 3D object.** At pre-training stage, The pre-training task we defined is to reconstruct the input mesh. Encoder and Decoder are trained at this stage. After the pre-training stage, we use SVM and MLP combined with the pre-train encoder to do the binary classification task.

where s and $\hat{s}$ are the ground truth and predicted point sets. M and N denote the number of points in the ground truth and predicted point sets. p_n and $\hat{p}_m$ are points in point set s and $\hat{s}$.

After the pre-training stage, we have two options to finish the binary classification task. The first option is to freeze the pre-trained model and use the pre-trained encoder to extract the features. After obtaining the features, we could use non-learning methods such as SVM to do the binary classification. In our work, we use SVM as one of our baselines to compare with our method. The second option is to open the trainable weights and train the network by loading the pre-trained model. A Multilayer Perceptron (MLP) is attached to the pre-trained encoder. We fine-tune the model by optimizing the cross-entropy loss function using stochastic gradient descent for 50 epochs. In all our experiments, we use the Adam optimizer with a fixed learning rate of 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Besides, the batch size is set to 4, and the number of faces per node is 8192.

2.3. Baselines

We use four baselines to compare with our method.

Volume After processing the meshes into watertight, we compute the volume of meshes. Then we combine Neighborhood Components Analysis (NCA) with k-nearest neighbors classifier to classify the features.

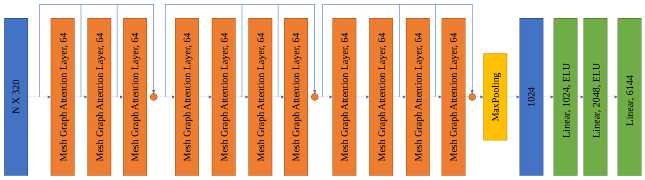
N X 320
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
Mesh Graph Attention Layer, 64
MaxPooling
1024
Linear, 1024, ELU
Linear, 2048, ELU
Linear, 6144

Fig. 3. **Neural Network Architecture.** N stands for the number of face nodes. The number after each layer stands for the dimension of the output embedding. The orange dot stands for concatenating the forward embeddings from previous layers. Each graph attention convolution layer has batch normalization and ReLU layer following.

Group	Total number	Training	Validation	Testing
AD Aβ+	151	96	24	31
MCI Aβ+	171	109	27	35
MCI Aβ-	171	109	27	35
CU Aβ+	116	74	18	24
CU Aβ-	232	148	37	47

Table 1. **The number of subjects contained in our database.**

PointNet++ Based on a pioneer work PointNet [12], PointNet++ [13] achieves better results on recognition of a point cloud than PointNet. To train PointNet++, we first extract 60,000 vertices on the hippocampus mesh. During training, we use Furthest Point Sampling to sample 2500 points on these vertices to feed into PointNet++.

SDF Sparse Convolution is first introduced in [14] which applies sparse convolution on the signed distance field of brain biomarkers. We first random sample 250000 points with signed distance information and sample 10000 points to feed into the neural network.

Support Vector Machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. We apply SVM to the features extracted from our pre-trained encoder.

3. EXPERIMENTS

3.1. Subjects

We use the ADNI database [15] for testing the performance of baselines and our algorithm. We divide the combined database into five classes according to the clinical groups and the positivity of amyloid biomarkers. The number of each class and the number of training, validation, and test subjects are listed in Table 1. We split the data set with a ratio of 80% for training and 20% for testing. We further split the training data into 80% to train the network and 20% for validation. For pre-training, we withhold the test data from the model to train. In the training process, the lable information is never used. After the pre-training, the same division is applied to train, validate, and test SVM and MLP.

We extract and reconstruct hippocampal surfaces using our brain subcortical morphometry analysis tool [16]. A centiloid cutoff of 37.1 is used to determine Aβ positivity, this threshold corresponds to pathologically determined moderate to frequent plaques [17]. We process hippocampal meshes by first making them watertight. This step is critical for analyzing meshes. Once the meshes are watertight, the graph computed from the mesh is well-connected, which means that two nodes that are close on the manifold are also close on the graph and vice versa. The second step is to simplify the mesh. In Fig. 4, we show the meshes of one subject under different resolutions.

Group	AD Aβ+ vs CU Aβ-					MCI Aβ+ vs MCI Aβ-					CU Aβ+ vs CU Aβ-				
Fold	V	P++	SDF SC	SVM	**MLP**	V	P++	SDF SC	SVM	**MLP**	V	P++	SDF SC	SVM	**MLP**
0	76.9	85.2/**78.2**	83.6/76.9	75.6	82.0/76.9	51.4	72.2/57.1	68.5/48.6	59.7	74.1/**62.9**	66.2	69.1/60.6	67.3/66.2	**67.1**	69.1/66.2
1	71.8	85.2/**80.8**	86.9/71.8	78.9	80.3/75.6	44.3	63.0/61.4	72.2/54.3	58.8	64.8/**67.1**	54.9	71.0/64.8	69.1/66.2	66.2	70.9/**69.0**
2	70.5	85.2/69.2	86.9/**78.2**	74.0	77.0/**78.2**	52.9	70.9/50.0	61.1/**61.4**	53.0	68.5/**61.4**	62.0	67.3/66.2	67.3/66.2	66.7	70.9/**69.0**
3	71.8	82.0/76.9	78.7/**82.1**	76.3	80.3/78.2	45.7	75.9/55.7	66.7/57.1	44.3	70.4/**58.6**	60.6	67.3/**66.2**	69.1/63.4	65.2	69.1/**66.2**
4	70.5	86.9/70.5	85.2/75.6	70.1	85.2/**78.2**	42.9	72.2/52.9	66.7/54.3	54.4	77.8/**67.1**	64.8	72.7/62.0	67.3/**66.2**	50.7	76.4/59.2
5	66.7	86.9/80.8	83.6/76.6	80.3	83.6/**83.3**	52.9	63.0/61.4	61.1/51.4	58.2	68.5/**65.7**	60.6	67.3/**66.2**	70.9/**66.2**	65.2	70.9/64.8
6	67.9	83.6/**79.5**	78.7/76.9	78.9	78.7/**79.5**	45.7	79.6/**64.3**	72.2/54.3	60.3	77.8/62.9	60.6	70.9/63.4	67.3/66.2	66.2	72.7/**67.6**
7	69.2	86.9/**80.8**	86.9/79.5	72.4	83.6/70.5	45.7	66.7/61.4	66.7/61.4	61.4	63.0/**67.1**	57.7	76.4/60.6	69.1/64.8	67.2	72.7/**69.0**
8	70.5	85.2/71.8	82.0/74.4	84.6	83.6/**84.6**	40.0	72.2/**65.7**	68.5/58.6	50.7	70.4/62.9	63.4	69.1/60.6	67.3/66.2	68.1	76.4/**69.0**
9	66.7	82.0/76.9	77.0/**79.5**	68.0	82.0/73.1	54.3	61.1/**65.7**	59.3/62.9	62.3	59.3/**65.7**	63.4	70.9/57.7	67.3/**66.2**	65.7	72.7/56.3
m/s	70.3±2.9	76.5±4.2	77.2±2.7	75.9±4.7	**77.8±4.0**	47.6±4.7	59.6±5.2	56.4±4.4	56.3±5.4	**64.1±2.7**	61.4±3.2	62.8±2.8	**65.8±0.9**	64.8±4.8	65.6±4.2

Table 2. **The comparison among different groups.** V represents the method using volume (detail in section 2.3). P++ represents PointNet++ (detail in section 2.3) and SDF SC is SDF Sparse Convolution (detail in section 2.3). SVM (section 2.3) and MLP (section 2.2) are our results with the proposed pre-training and fine-turning framework.

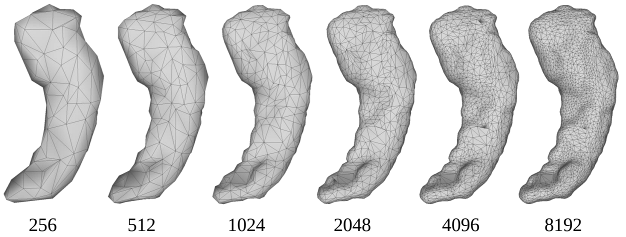

Fig. 4. **Visual comparisons among different number of faces of hippocampal surfaces.** The number means the number of faces in the mesh shown in the figure.

When the number of faces is too small, we lose the detailed information which may be important for the classification. However, a large number of faces could make the GPU out of memory. Thus to balance the trade-off between fidelity and computation viability, we choose 8192 as the number of faces we simplified every subject to.

3.2. Improvement Over the Baselines

In Table 2, we compare the accuracy over three groups and five methods. The three groups are AD Aβ+ vs CU Aβ-, MCI Aβ+ vs MCI Aβ-, and CU Aβ+ vs CU Aβ-. We split the data set into 10 folds and take each group's average test accuracy of each method. Our proposed method outperforms 19 folds out of 30 folds over other methods. As a comparison, PointNet++ and SDF Sparse Convolution outperform 9 and 7 folds out of 30 folds. Our method achieves the best average test accuracy of 77.8 in the AD Aβ+ group vs. CU Aβ-. In the MCI Aβ+ vs. MCI Aβ- group, our method gets average test accuracy of 64.1, which is an 8% improvement compared with previous methods (P++). In the CU Aβ+ vs. CU Aβ- group, our method gets an average test accuracy of 65.6, which is 0.2% lower than a previous method (SDF SC).

PointNet++	SDF Sparse Convolution	**MLP**
	AD Aβ+ vs CU Aβ-	
8.4	5.8	**3.8**
	MCI Aβ+ vs MCI Aβ-	
10.1	9.9	**5.3**
	CU Aβ+ vs CU Aβ-	
7.4	**2.4**	6.6

Table 3. **Difference between validation accuracy and test accuracy.** The lower means models are less over-fitting.

In Table 3, we further compute the difference between average and test validation accuracy. Our method achieves the lowest differences in two groups, AD Aβ+ vs. CU Aβ- and MCI Aβ+ vs. MCI Aβ-. The difference between validation and test metric is an important sign in measuring the robustness and generalizability of a trained model. Our method obtains a low difference, which means that pre-training and fine-tuning in our setting leads the model to a better local minimum than directly training models for classification tasks. The results also show that our method is less over-fitting than previous methods. Considering the relatively small size of the dataset (in some cases, obtaining labels is time-consuming and difficult in many aspects), pre-training provides a direction to train the model on a large number of unlabeled data and then uses the pre-trained model for the downstream tasks. In our case, we use the entire training dataset to pre-train the model (without using the label information) and fine-tune the model on specific tasks (using the label information). It shows the viability and feasibility of applying pre-training and fine-tuning procedures in the biomedical domain.

4. CONCLUSION AND FUTURE WORK

In summary, our work is the first to use a graph-attention-based deep neural network combined with pre-training and fine-tuning to identify brain amyloid burdens. Graph attention convolution and pre-training provide a new way of analyzing brain images. Our method exhibits the robustness and generalizability of a trained model compared with directly training models for classification tasks. With more data being collected, the gap between direct classification and pre-training could become larger, and the performance drop between validation and test accuracy could be smaller.

5. COMPLIANCE WITH ETHICAL STANDARDS

This research study was conducted retrospectively using human subject data available in open access by [15]. Ethical approval was not required as confirmed by the license attached with the open-access data.

6. ACKNOWLEDGMENTS

This work was partially supported by the National Institutes of Health (R01EB025032, R01DE030286, R01EY032125, R21AG065942, R01AG069453, RF1AG073424, P30AG072980), and the Arizona Alzheimer Consortium.

7. REFERENCES

[1] Ron Brookmeyer, Elizabeth Johnson, Kathryn Ziegler-Graham, and H Michael Arrighi, "Forecasting the global burden of alzheimer's disease," *Alzheimer's & dementia*, vol. 3, no. 3, pp. 186–191, 2007.

[2] C. Salvatore, A. Cerasa, P. Battista, M. C. Gilardi, A. Quattrone, and I. Castiglioni, "Magnetic resonance imaging biomarkers for the early diagnosis of Alzheimer's disease: a machine learning approach," *Front Neurosci*, vol. 9, pp. 307, 2015.

[3] Bradley T Hyman, "Amyloid-dependent and amyloid-independent stages of alzheimer disease," *Archives of neurology*, vol. 68, no. 8, pp. 1062–1064, 2011.

[4] R. J. Bateman, K. Blennow, R. Doody, S. Hendrix, S. Lovestone, S. Salloway, R. Schindler, M. Weiner, H. Zetterberg, P. Aisen, and B. Vellas, "Plasma Biomarkers of AD Emerging as Essential Tools for Drug Development: An EU/US CTAD Task Force Report," *J Prev Alzheimers Dis*, vol. 6, no. 3, pp. 169–173, 2019.

[5] S. Janelidze, N. Mattsson, S. Palmqvist, R. Smith, T. G. Beach, G. E. Serrano, X. Chai, N. K. Proctor, U. Eichenlaub, H. Zetterberg, and et al., "Plasma P-tau181 in Alzheimer's disease: relationship to other biomarkers, differential diagnosis, neuropathology and longitudinal progression to Alzheimer's dementia," *Nat Med*, vol. 26, no. 3, pp. 379–386, 03 2020.

[6] J. Wu, Q. Dong, J. Gui, J. Zhang, Y. Su, K. Chen, P. M. Thompson, R. J. Caselli, E. M. Reiman, J. Ye, and Y. Wang, "Predicting Brain Amyloid Using Multivariate Morphometry Statistics, Sparse Coding, and Correntropy: Validation in 1,101 Individuals From the ADNI and OASIS Databases," *Front Neurosci*, vol. 15, pp. 669595, 2021.

[7] Kyle Strimbu and Jorge A Tavel, "What are biomarkers?," *Current Opinion in HIV and AIDS*, vol. 5, no. 6, pp. 463, 2010.

[8] L. G. Apostolova, J. H. Morra, A. E. Green, K. S. Hwang, C. Avedissian, E. Woo, J. L. Cummings, A. W. Toga, C. R. Jack, M. W. Weiner, and P. M. Thompson, "Automated 3D mapping of baseline and 12-month associations between three verbal memory measures and hippocampal atrophy in 490 ADNI subjects," *Neuroimage*, vol. 51, no. 1, pp. 488–499, May 2010.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016.

[12] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *CoRR*, vol. abs/1612.00593, 2016.

[13] C Qi, L Yi, H Su, and L Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017.

[14] Z. Yang, J. Wu, P. M. Thompson, and Y. Wang, "Deep Learning on SDF for Classifying Brain Biomarkers," *Annu Int Conf IEEE Eng Med Biol Soc*, vol. 2021, pp. 1051–1054, 11 2021.

[15] Ronald Carl Petersen, PS Aisen, Laurel A Beckett, MC Donohue, AC Gamst, Danielle J Harvey, CR Jack, WJ Jagust, LM Shaw, AW Toga, et al., "Alzheimer's disease neuroimaging initiative (adni): clinical characterization," *Neurology*, vol. 74, no. 3, pp. 201–209, 2010.

[16] J. Shi, P. M. Thompson, B. Gutman, and Y. Wang, "Surface fluid registration of conformal representation: application to detect disease burden and genetic influence on hippocampus," *Neuroimage*, vol. 78, pp. 111–134, Sep 2013.

[17] A. S. Fleisher, K. Chen, X. Liu, A. Roontiva, P. Thiyyagura, N. Ayutyanont, A. D. Joshi, C. M. Clark, M. A. Mintun, M. J. Pontecorvo, and et al., "Using positron emission tomography and florbetapir F18 to image cortical amyloid in patients with mild cognitive impairment or dementia due to Alzheimer disease," *Arch Neurol*, vol. 68, no. 11, pp. 1404–1411, Nov 2011.