Brain-inspired Approaches for Advancing Artificial Intelligence

by

Haixing Dai

(Under the Direction of Tianming Liu)

Abstract

Deep learning has experienced rapid growth and garnered significant attention in recent decades. Simultaneously, neuroscience has remained a challenging and enigmatic field of study. Inspired by the structure and function of the brain, researchers have developed increasingly powerful and sophisticated deep learning models that have achieved remarkable performance in various domains, including computer vision, natural language processing, and medical image analysis. These brain-inspired models have revolutionized the field of artificial intelligence, enabling breakthroughs in tasks such as image recognition, language understanding, and disease diagnosis. In turn, the application of these advanced deep learning models has provided valuable insights into the inner workings of the human brain, revealing temporal and spatial functional brain networks. The symbiotic relationship between artificial intelligence and neuroscience is evident, as they continuously inform and complement each other's progress.

This dissertation presents novel frameworks that integrate deep learning and knowledge from brain science. This research aims to gain insights into the brain and refine deep learning models through brain-inspired principles. The dissertation first discusses how deep learning has been applied to study the brain, focusing on areas such as modeling cortical folding patterns, hierarchical brain structures, and spatial-temporal brain networks. It then discusses how artificial neural networks have drawn inspiration from the brain, using examples like convolutional neural networks, attention mechanisms, and language models. The dissertation's main contributions are several computational frameworks integrating brain-inspired insights. These include a graph representation neural architecture search method to optimize recurrent neural networks for analyzing spatiotemporal brain networks, a hierarchical semantic tree concept whitening model to disentangle concept representations for image classification, a twin-transformer framework to study gyri and sulci in the cortex, a core-periphery guided vision transformer, and methods leveraging language models to generate data and analyze health narratives. Overall, this dissertation explores how we can understand the brain better using deep learning and ultimately build more efficient, robust, and interpretable artificial neural networks inspired by the brain.

Index words: Deep Learning, Brain-inspired AI, Large Language Model, Brain Neural Network, Artificial Neural Network, Casual Inference, Medical Image Analysis.

Brain-inspired Approaches for Advancing Artificial Intelligence

by

Haixing Dai

B.S., University of Electronic Science and Technology of China, China, 2016

A Dissertation Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the
Degree.

Doctor of Philosophy

Athens, Georgia

2023

Brain-inspired Approaches for Advancing Artificial Intelligence

by

Haixing Dai

Major Professor:	Tianming Liu
Committee:	Sheng Li Ninghao Liu

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
August 2023

Dedication

This dissertation is a heartfelt dedication to my parents, whose boundless love and unwavering support have served as a constant source of strength throughout my life and academic journey. I am eternally grateful for their sacrifices, guidance, and unwavering encouragement, as they have played a significant role in shaping the person I am today. Additionally, I would like to extend this dedication to my girlfriend, Xin Ye, whose love, companionship, and unwavering support have been instrumental during my Ph.D. studies. Her presence has brought joy and stability, and I am deeply appreciative of her unwavering belief in me. I would also like to express my profound appreciation to my four delightful furry friends who have brought immeasurable joy to my journey as a PhD student. To Pebbles, Pixie (the daughter of Pebbles), Viola, and my couch potato companion Lucas, I am endlessly grateful for the boundless fun and happiness they have bestowed upon me. Their unwavering companionship and unconditional love have been a constant source of comfort and inspiration throughout my career. Thanks for being my steadfast companions and bringing warmth to every moment.

Acknowledgments

I wish to express my deepest appreciation to Professor Tianming Liu for his indispensable mentorship, backing, and motivation throughout my journey towards earning my doctorate. The depth of Prof. Liu's wisdom, his astute observations, and his productive feedback have been instrumental in molding my research and fostering my development as a scholar. His fervor for investigation and his commitment to his pupils are genuinely motivating. I consider myself privileged to have had the chance to evolve under his tutelage in both scholarly pursuits and life over the preceding half-decade. I shall forever be thankful for the wisdom and abilities he has instilled in me, the resources and chances he facilitated, and everything he has contributed to my development.

My heartfelt appreciation goes out to my Ph.D. advisory committee members, Dr. Sheng Li and Dr. Ninghao Liu, for their invaluable inputs and perceptive remarks throughout my doctorate voyage. Their constructive critique, motivation, and backing have guided me through the challenges I've encountered on my path. I am both honored and fortunate to have had them on my Ph.D. advisory committee, and their mentorship and companionship will forever hold a special place in my heart.

I would like to express my heartfelt gratitude to Dr. Dajiang Zhu, Dr. Xi Jiang, Dr. Gengcheng Mai, Dr. Dinggang Shen, Dr. Quanzheng Li and Dr. Xiang Li for their invaluable guidance, encouragement, and insightful feedback throughout my doctoral journey. Their expertise and mentorship have played a pivotal role in shaping me as a researcher and scholar, and I am deeply grateful for their unwavering support. Their dedication to academic excellence and their willingness to share their knowledge have been truly inspiring. I would like to express my sincere appreciation to my colleagues, both within and outside our laboratory, for their invaluable collaboration, unwavering camaraderie, and continuous support. Their contributions have played a significant role in the success of this project. Special thanks go to Lu Zhang, XiaoweiYu, Lin Zhao, Zhengliang Liu, Zihao Wu, Qing Li, Mengxuan Hu, Wenxiong Liao, Chong Ma, Yiwei Li, Wei Zhang, Yu Zhao, Fangfei Ge, Peijun Wu and Ming Gong for their exceptional dedication and teamwork. The stimulating discussions, shared experiences, and collective efforts have enriched my doctoral experience and contributed to the successful completion of this dissertation. I am grateful for the friendships forged and the sense of community we have fostered together. My deepest thanks go to all those who have contributed to my journey, as your contributions have been instrumental in my growth and accomplishments.

Specifically, I would like to express my heartfelt gratitude to the mentors who played a crucial role during my internships at TikTok and Google. I am truly grateful to Jun Xiao, Deguang Kong, Wenchao Zheng, Shuang Liu, Leon Lin, and Tung Nguyen for their selfless dedication in sharing their expertise in both academic and industrial fields. Their guidance, insights, and support have been invaluable in shaping my understanding and skills. The mentorship I received from them has not only contributed to the success of my internships but will also serve as a solid foundation for my future endeavors. I am confident that the knowledge and experiences gained under their mentorship will enable me to make significant contributions to society in the years to come. Their mentorship has been an invaluable gift, and I am truly grateful for their guidance and support throughout my internships.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation delves into the intersection of deep learning and brain science, with a special focus on brain-inspired artificial intelligence (AI). Our research aims to better understand the brain and further refine deep learning models through brain-inspired principles and findings. We cover diverse topics in this research direction and explore the potential of brain-inspired AI.

1.1 Deep Learning for Neuroscience

Deep learning has wide applications in various domains, including neuroscience (Kellmeyer, 2019; Marblestone et al., 2016). Recent developments in deep learning are capable of recognizing patterns and deciphering unstructured data (L. Zhao, Zhang, et al., 2023). Since the rise of deep learning in the 2010s (Schmidhuber & Blog, 2020), there has been significant efforts to apply latest deep learning methods to the study of the brain (L. Zhang, Wang, et al., 2020; L. Zhao, Zhang, et al., 2023).

In studying brain folding patterns, a significant piece of research proposed an innovative 'cortex2vector' framework (L. Zhang et al., 2023). This project tackled the encoding of individual cortical folding patterns into anatomically meaningful embedding vectors, providing a unique means to represent the nuanced structure of the brain. For example, this study utilized a learning-based framework to translate the complex folds of the brain into a mathematical representation that can be better understood and compared across different brains.

Another study in this field developed a topology-preserving transfer learning framework to differentiate fMRI time series derived from cortical folds (S. Liu et al., 2022). This is particularly crucial for understanding differences between Autism Spectrum Disorder (ASD) and healthy controls. By preserving the topological structure of the brain during analysis, this research was able to isolate unique structural characteristics associated with ASD, providing novel insights into its diagnosis and treatment.

In the domain of exploring and modeling hierarchical structures within the brain, various research have made significant strides (Pang et al., 2022). For instance, one study devised an unsupervised differentiable neural architecture search algorithm. This research aimed at automating the design of deep belief networks, a type of neural network optimized for hierarchical functional brain network decomposition. By using a unique Gumbel-Softmax scheme, they managed to reframe the discrete architecture sampling procedure into a continuous process, providing more accurate modeling of the brain's hierarchical structure.

Another important study in this domain proposed a multimodal deep belief network model (S. Zhang et al., 2019), which has the capability to discover and represent the hierarchical organizations of common and consistent brain networks from both fMRI and DTI data. This research went beyond the traditional means of neuroimaging data analysis by integrating different modalities, thereby enhancing the understanding of the brain's architecture.

In the area of temporal and spatial pattern representation, various studies employed different neural network architectures. One used a deep convolutional autoencoder to learn mid-level and high-level features from complex, large-scale task-based fMRI time series in an unsupervised manner (H. Huang et al., 2017). Another proposed a deep sparse recurrent autoencoder for simultaneous extraction of spatial patterns and temporal fluctuations of brain networks (Q. Li et al., 2019). Yet another study made use of an unsupervised embedding framework based on Transformer to encode brain function into dense vectors (L. Zhao et al., 2022), paving the way for more accurate representation of the dynamic aspects of brain function.

Spatial-temporal modeling of functional brain networks is another area of intense research focus. For example, one study delved into fully Bayesian spatio-temporal modeling of fMRI data (Woolrich et al., 2004). They offered a new means of analyzing fMRI data, incorporating a high level of statistical rigor into the model. Meanwhile, another study presented a two-stage deep learning framework (Y. Zhao et al., 2019), offering a comprehensive and systematic approach to spatial-temporal resting state network modeling.

When it comes to the exploration of the core-periphery structure in gyri and sulci, one particular study is worth mentioning (X. Yu, Zhang, Dai, Zhao, et al., 2023). This work presented a unique Twin-Transformer framework to delve into the unique functional roles of gyri and sulci, as well as their interactions. This study took the innovative step of using separate transformers to process information from gyri and sulci, revealing a new layer of complexity in how these two structures interact.

Lastly, in the domain of multimodality fMRI studies, researchers have sought to couple various types of data with brain function. For instance, one study coupled the visual semantics of artificial neural networks and human brain function via synchronized activations (L. Zhao, Dai, et al., 2023), creating a more comprehensive model of how the brain processes visual information. Another study aimed to link the neurons in the popular natural language processing model BERT with the biological neurons in the human brain (X. Liu et al., 2023), providing an entirely new perspective on natural language understanding. Yet another study proposed a brain-inspired adversarial visual attention network to characterize human visual attention directly from functional brain activity (H. Huang et al., 2022). This research integrated multiple sources of data to create a more nuanced understanding of how attention works in the human visual system. The BI-AVAN framework characterizes human visual attention directly from functional brain activity, unlike many previous studies which relied primarily on eye-tracking data. It simulates the competitive dynamics between attention-related and attention-neglected objects, successfully identifying and locating the visual objects that the human brain focuses on in an unsupervised manner. By utilizing independent eye-tracking data as validation, we demonstrated that our model offers robust and promising results in discerning meaningful human visual attention and mapping the relationship between brain activities and visual stimuli.

1.2 Refining Deep Learning through Brain-Inspired AI

In this section, we delve into how artificial neural networks draw inspiration from biological neural networks, as well as our own contributions to brain-inspired artificial intelligence.

From the inception of artificial neural networks (ANNs), biological neural networks (BNNs) have been a rich source of inspiration. For example, the design of convolutional neural networks (CNNs) (LeCun, Bengio, et al., 1995), a core component in many deep learning models, was motivated by the cat's visual cortex (Hubel & Wiesel, 1962). The visual cortex was found to have neurons that individually responded to small regions of the visual field, forming a local receptive field. These discoveries led to the concept of local receptive fields in CNNs, where individual neurons process data for a specific region of the image.

Similarly, the attention mechanism of the transformer architecture (Vaswani et al., 2017), a key component in many state-of-the-art natural language processing models, also mirrors mechanisms in the human brain. The attention mechanism was designed to weight the importance of different inputs, similar to how the human brain prioritizes information.

Moreover, BERT (Devlin et al., 2018), a groundbreaking model for natural language understanding, and the Vision Transformer (ViT) (Dosovitskiy et al., 2020), a revolutionary model in the field of computer vision, both exhibit architectures reminiscent of aspects of the brain. BERT's attention mechanism parallels the brain's capacity to focus on specific parts of the linguistic input when constructing meaning, and ViT's architecture parallels the brain's visual processing system, with its emphasis on capturing global, contextual information.

Turning to our own contributions, our work on Hierarchical Semantic Tree Concept Whitening (HaST-CW) attempts to disentangle concepts with hierarchical relations in the context of image classification. This method was inspired by the way the human brain processes hierarchical information, allowing us to improve the interpretability and robustness of the model by forcing it to decorrelate the latent representations of different concepts.

Our Core-Periphery Principle Guided Redesign of Self-Attention in Transformers (CP-ViT) (X. Yu, Zhang, Dai, Lyu, et al., 2023) used the core-periphery principle, a common organizational paradigm in human brain networks, to enhance the performance and interpretability of ViTs. We designed a sparse graph guided by the core-periphery structure, enabling more efficient and meaningful information exchange in the self-attention mechanism.

Inspired by the human ability to describe various concepts using different words and sentence structures while preserving their underlying meanings, large language models such as ChatGPT and GPT4 exhibit a similar capacity. We employed large language models such as ChatGPT, which we believe can approximate certain aspects of the human brain's complexity and capacity for generalization, in our work on AugGPT. Here, we leveraged these models to generate auxiliary samples for few-shot text classification, showcasing the power of these models in handling sparse data scenarios, much like the human brain can learn from a few examples.

Lastly, in our AD-AutoGPT project, we were inspired by the brain's ability to break down complex tasks into multiple subtasks, execute them, and integrate the results. We developed an automated system capable of collecting, processing, and analyzing complex health narratives of Alzheimer's Disease based on users' textual prompts. This project underscores how we can learn from the brain's operational principles to design more effective and autonomous systems.

In sum, the brain continues to inspire the design and operation of artificial neural networks, guiding us towards more efficient, robust, and interpretable models. By observing and understanding the mechanisms of our own brains, we can make strides towards creating artificial general intelligence, and our research contributes to this exciting journey.

1.3 Contributions

To address these challenges and explore brain-inspired AI, we proposed a series of computational frameworks to show 1) how we leverage the deep learning model to understand the function brain networks. 2) how we optimize the deep learning model under the direction of brain-inspired pattern. The contributions of this dissertation are summarized as follows:

- We proposed a novel graph representation neural architecture search (GR-NAS) method based on graph representation to optimize the vanilla RNN cell structure for decomposing spatial/temporal brain networks.

- We utilized a novel graph representation-based neural architecture search (GR-NAS) model to optimize the inner cell architecture of recurrent neural network (RNN) for decomposing the spatio-temporal FBNs and identifying the neuroimaging biomarkers of subtypes of PAE.

- We proposed a novel Hierarchical Semantic Tree Concept Whitening (HaST-CW) model to decorrelate the latent representations in image classification for disentangling concepts with hierarchical relations.

- We introduced a novel Twin-Transformer to represent and unveil the fundamental functional roles of the two basic cortical folding patterns: gyri and sulci.

- We leveraged the Core-Periphery (CP) organization, which is widely found in human brain networks, to guide the information communication mechanism in the self-attention of vision transformer (ViT) and name this novel framework as CP-ViT.

- We proposed a graph varying coefficient neural network (GVCNet) for estimating the individual treatment effect with continuous treatment levels using a graph convolutional neural network.

- We propose a new data augmentation method named AugGPT, which leverages ChatGPT to generate auxiliary samples for few-shot text classification.

- Inspired by AutoGPT, the state-of-the-art open-source application based on the GPT-4 large language model, we develop a novel tool called AD-AutoGPT which can conduct data collection, processing, and analysis about complex health narratives of Alzheimer's Disease in an autonomous manner via users' textual prompts.

Overall, this dissertation mainly focus on 1) the deep learning model used for exploring the function brain network on Medical Image and 2) Optimizing the deep learning model based on the inspiration of structure and function of brain neural networks.

1.4 Dissertation Outline

This dissertation contains 8 chapters.

Chapter 2 introduces the investigation of spatial/temporal Function Brain Networks with a novel graph representation neural architecture search (GR-NAS) model. First, The novel model has significantly well performance on optimally decomposing spatial/temporal functional brain networks from fMRI data. Second, GR-NAS was used to identify the neuroimaging biomarkers of Prenatal alcohol exposure (PAE) groups, whcih provides a new perspective for the abnormal brain early diagnosis with fMRI data.

Chapter 3 details the novel Hierarchical Semantic Tree Concept Whitening (Hast-CW) model. Rather than relying on post-hoc schemes, we proactively instill knowledge to alter the representation of human-understandable concepts in hidden layers. Specifically, we use a hierarchical tree of semantic concepts to store the knowledge, which is leveraged to regularize the representations of image data instances while training deep models.

Chapter 4 explores the brain's basic structural and functional mechanisms with a novel Twin-Transformer, then leverage the found core-perphery pattern to guide the vision transformer (ViT). In the first study, A novel Twin-Transformer framework was designed to explore and unveil the unique functional roles of gyri and sulci as well as their relationship and interaction in the whole brain function. Second, inspired by the first study, we utilized the principles found in BNNs to guide and improve our ANN architecture design.

Chapter 5 introduces the novel GVCNet model for measuring the regional causal connections between amyloid-β accumulation and AD pathophysiology, which may serve as a robust tool for early diagnosis and tailored care.

Chapter 6 introduces a text data augmentation approach based on ChatGPT (named AugGPT). AugGPT rephrases each sentence in the training samples into multiple conceptually similar but semantically different samples for downstream model training.

Chapter 7 details the novel tool called AD-AutoGPT which can conduct data collection, processing, and analysis about complex health narratives of Alzheimer's Disease in an autonomous manner via users' textual prompts.

Chapter 8 concludes the whole dissertation and discusses future works.

Chapter 2

Neural Architecture Search in Medical Image AI

2.1 Graph Representation Neural Architecture Search for Optimal Spatial/Temporal Functional Brain Network Decomposition

2.1.1 Overview

Decomposing the spatial/temporal functional brain networks from 4D functional magnetic resonance imaging (fMRI) data has attracted extensive attention. Among all these efforts, deep neural network-based methods have shown significant advantages due to their powerful hierarchical representation ability. However, the network architectures of those deep learning models are manually crafted, which is time consuming and non-optimal. This paper presents a novel graph representation neural architecture search (GR-NAS) method based on graph representation to optimize the vanilla RNN cell structure for decomposing spatial/temporal brain networks. The core idea is to embed the discrete search space of the RNN cell into a continuous domain that preserves the topological information. After that, popular search algorithms, e.g., reinforcement learning (RL) and Bayesian optimization (BO), can be employed to find the optimal architecture in this continuous space. The proposed method was evaluated on the Human Connectome Project (HCP) task fMRI datasets. Extensive experiments demonstrated the superiority of the proposed model in brain network decomposition both spatially and temporally. To our best knowledge, the proposed model is among the early efforts using NAS strategy to optimally decompose spatial/temporal functional brain networks from fMRI data.

2.1.2 Background

Exploring functional brain networks (FBNs) has been an active research topic in neuroimaging field for years (Woolrich et al., 2004; Y. Zhao et al., 2019). To decompose the FBNs from functional magnetic resonance imaging (fMRI) data, various model-driven and data-driven methods have been proposed (X. Hu et al., 2018; H. Huang et al., 2017; Q. Li et al., 2019; H. Wang et al., 2018; W. Zhang et al., 2019; W. Zhang et al., 2020), among which deep learning-based approaches have gained increasing attention because of their powerful and hierarchical representation ability. For example, a deep convolutional auto-encoder (DCAE) was proposed to explore both the high-level and low-level FBNs (H. Huang et al., 2017). Another group of studies employed a deep belief network (DBN) to identify the FBNs in a hierarchical manner (W. Zhang et al., 2019; W. Zhang et al., 2020). Recently, recurrent neural network (RNN) based models have shown significant advantages in modeling temporal dependencies within fMRI data and achieved promising results (Q. Li et al., 2019; H. Wang et al., 2018). For instance, a deep sparse recurrent autoencoder (DSRAE) (Q. Li et al., 2019) was developed to simultaneously decompose the FBNs at connectome-scale, demonstrating the effectiveness of RNN models in extracting neuroscientifically meaningful spatial/temporal networks from 4D fMRI data.

However, current deep learning models such as the abovementioned DSRAE are limited in the sense that their network architectures are manually crafted. Generally, designing an appropriate or optimal neural network is a laborious and time-consuming process that greatly depends on rich domain knowledge and experience. Also, for modeling the FBNs under different task stimuli, the neural network architectures might need to be optimized, respectively. For example, the optimal hyper-parameters of DSRAE for the emotion task might not be optimal for the working memory task process. To overcome these challenges, some literature efforts were made based on evolutionary neural architecture search (NAS) to find the optimal architectures for 4D fMRI data (Q. Li et al., 2020; Yan et al., 2020). Another literature work that adopted differentiable neural architecture search (DARTS) (H. Liu et al., 2018), named ST-DARTS (Q. Li et al., 2021), was proposed to improve the efficiency of searching optimal RNN architectures while maintaining comparable performances. Nevertheless, ST-DARTS is still limited in discrete space and neglects the topological information within RNN cells, which might be trapped on a local optimum and thus degenerates the performance.

This paper proposed a novel graph representation NAS (GR-NAS) method to optimize the vanilla RNN cell structure for decomposing the spatial/temporal FBNs. Specifically, we represented the RNN cell architectures in the discrete DARTS search space as graphs and embedded them into a latent continuous search space via a Graph Isomorphism Network (GIN) (K. Xu et al., 2018) encoder. With the embeddings in this continues space, several search strategies, such as reinforcement learning (RL) (Zoph & Le, 2016) and Bayesian optimization (BO) (Williams, 1992), can be adopted to search for the optimal RNN cells. We evaluated the proposed unsupervised NAS framework on publicly available Human Connectome Project (HCP) task fMRI data with RL and BO search strategies. Extensive experimental results demonstrated the superiority of the proposed method in both searching optimal network architectures and decomposing meaningful FBNs. Moreover, the transfer learning process based on unsupervised NAS between different tasks can also achieve good performance, suggesting the robustness and generality of the proposed method.

2.1.3 Materials and Method

Overview

Fig. 2.1 presents the overview of our proposed GR-NAS model. The general target of this work is to learn the optimal RNN cells for decomposing the spatial/temporal FBNs from fMRI data. RNN cells, which have graph-structured architectures, are embedded with the Graph Isomorphism Autoencoder to learn the pre-trained embeddings. Then, two different downstream architecture search algorithms are adopted to search the optimal architecture on the pre-trained embeddings. Finally, we apply the learned RNN cell to 4D fMRI data to obtain the feature map in the latent layer for decomposing the spatial/temporal brain function network.

Data Description and Pre-processing

We adopted the publicly available HCP grayordinate-based tfMRI datasets from the Q3 release of the Human Connectome Project (https://db.humanconnectome.org). The detailed acquisition parameters are as follows: TR=720ms, TE=33.1ms, flip angle=52°, in-plane FOV=208mm×180mm, 104×90 matrix, slice thickness=2mm, 72 slices, multiband factor=8, echo spacing=0.58ms, BW=2290Hz/Px. Important preprocessing steps including spatial smoothing, temporal filtering, nuisance regression, and motion censoring were applied to all subjects. In addition, we extracted the 4D fMRI volume of each subject and rearranged it into a 2D signal matrix for Emotion and Working Memory (WM) task, respectively. In

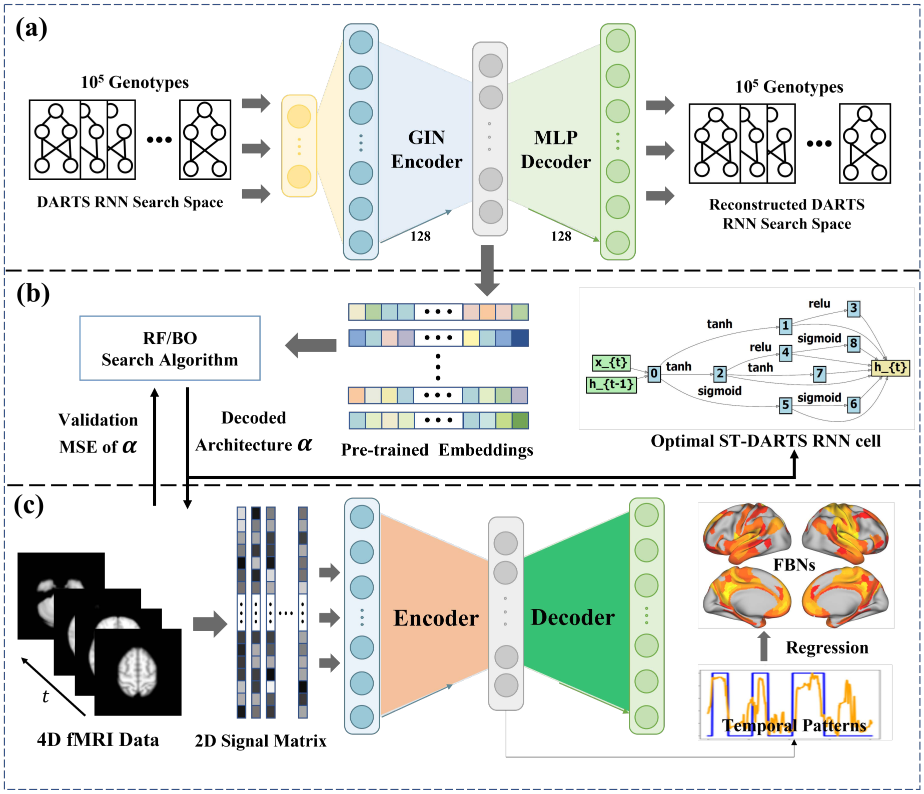


Figure 2.1: Our framework of GR-NAS. (a) GIN encoder and multiple layer perceptron (MLP) decoder are used to embed the DARTS RNN cell architectures to obtain pre-trained embeddings. (b) RF/BO search algorithms are used on the pre-trained embeddings to search for the optimal architecture based on the architecture performance estimation. (c) Decomposed spatial/temporal function network with the learned architecture. The input is our 4D-fMRI data which consists of the spatial (360 region of interests) and temporal (length of fMRI tasks' time duration) information. The selected DARTS RNN cell is evaluated by cross-entropy loss.

the training stage, we randomly selected 750 subjects and divided them into three groups: 450 subjects as training set, 150 subjects as validation set and another 150 subjects as testing set. During the network architecture search process, we only used the training set and validation set. All subjects were included in evaluating the searched architectures.

Graph Representation Neural Architecture Search

In this work, we proposed a novel GR-NAS model for searching the optimal architecture of RNN cell, and it has been applied to decompose the FBNs. Specifically, GR-NAS has a variational graph isomorphism autoencoder that embeds the represented graph of RNN cell into a continuous searching space. Then, search algorithms can be used to explore the optimal architecture.

Variational Graph Isomorphism Autoencoder: We set the searching space of our RNN cell as that of DARTS. In this space, each RNN cell can be represented as a directed acyclic graph (DAG) G= (V, E), where V is the set of nodes and E is the set of edges. Each node is associated with one of K predefined operations. Therefore, the DAG can be represented by two matrices: one upper triangular adjacency matrix $A \in R^{N\times N}$ encoding the connections among N nodes and another one-hot operation matrix $X \in R^{N\times K}$ recording the operation in each node. Then, a two-layer GIN encoder is used to embed the adjacency matrix A and operation matrix X into a vector $Z \in R^{N}$. The encoder is defined as:

$$q(Z|X,A) = \prod_{i=1}^{n} q(z_i|X,A), with\ q(z_i|X,A) \sim \mathcal{N}(z_i|\mu_i, diag(\sigma^2)) \tag{2.1}$$

where μ, σ are the mean and the variance of approximation $q(z_i|X,A)$, respectively. Then we use a L-layer GIN to obtain the node embedding matrix H:

$$H^{(k)} = MLP^{k}\ ((i+\epsilon^{k}\ ) \cdot H^{(k-1)} + A \cdot H^{(k-1)}), k = 1, 2, \ldots, L, \tag{2.2}$$

where $H^{(0)} = X$, ϵ is a trainable bias, and MLP is a multi-layer perceptron with each layer a linear-batchnorm-ReLU triplet. We use $H^{(L)}$ to obtain the mean μ and the variance σ of the $q(Z|X,A)$. After the encoder, a generative model, which aims at reconstructing $\widehat{X}$, $\widehat{A}$ from the latent variable Z, is defined as:

$$P(\widehat{A}|Z) = \textstyle\prod_{i=1}^{N}\prod_{j=1}^{N} p(\widehat{A}_{ij}|z_i,z_j), with\ P(\widehat{A}_{ij} = 1|z_i,z_j) = \sigma(z_i^{T} z_j) \tag{2.3}$$

$$P(\widehat{X} = [k_1, \ldots, k_n]^{T}|Z) = \textstyle\prod_{i=1}^{N} P(\widehat{X} = k_i|z_i) = \prod_{i=1}^{N} softmax(WZ+b) \tag{2.4}$$

whereσ is the logistic sigmoid function and A_{ij} contains the elements of A. With the autoencoder, we maximize the lower bound L:

$$\mathcal{L} = E_{q(X,A)}[logp(A|Z)] - KL[q(Z|X,A)||p(Z), \qquad (2.5)$$

where we assume that given the latent variable Z, the adjacent matrix A and the operation matrix X are conditionally independent. In other words, $p(X, A|Z) = p(A|Z)p(X|Z)$. The term KL is the KL divergence, which measures the difference between the posterior distribution $q(\cdot)$ and the prior distribution $p(\cdot)$. We perform full-batch gradient descent and use the reparameterization scheme to inject random noise to the training layer, which has been proven to be effective on the regularization of neural networks. The loss function is optimized using mini-batch gradient descent over neural architectures.

Search Strategies: This paper uses two different representative search algorithms to evaluate our model on pre-trained embeddings: reinforcement learning (RL) and Bayesian optimization (BO). In GR-NAS (RL), the embeddings are agents, the validation loss for each embedding is used as the reward, a single layer long short-term memory (LSTM) is used as the controller, and the action is used as the movement on one of the embedding's 16 dimensions. We leveraged the pre-trained embedding pass to the policy LSTM network to evaluate the current state and then obtain the next action and state based on L^2 distance to minimize the reward, i.e., the validation loss of selected embedding. We used the Adam optimizer and set the learning rate as 0.01. The discount factor is set to 0.9, and the baseline value is set to 25. The estimated wall-clock time for each run is set to 20 mins. In GR-NAS (BO), deep networks are used for global optimization (DNGO) (White et al., 2021) to search the optimal architecture on our embeddings. A one-layer adaptive basis regression network with 128 hidden dimensions is employed for modeling the distribution over function. During the training, selected pre-trained embeddings will receive corresponding loss values and then pass them to DNGO for selecting top architectures.

Temporal/Spatial Functional Network Learning

After GR-NAS searching on the embeddings, optimal RNN cells are found for spatial/temporal function network learning. The temporal network dynamics were learned with the latent RNN cells and the spatial networks were derived by such temporal network dynamics with Elastic Net regression (Zou & Hastie, 2005). Then we put these architectures into our temporal function network learning model, which is the same as the model used for the performance estimate strategy in Fig. 2.1(c). In the process of architecture searching, we set the epoch as 10. In the temporal function network learning, the epoch was set as 200 to guarantee the convergence with 128 batch size. For other hyperparameters, each cell has 8 nodes, and the number of operations is 4 (tanh, identity, sigmoid, ReLU). The hidden size is set as 32, and the initial learning rate is set as 20. We implemented the proposed GR-NAS model with PyTorch 1.4.1 on a single RTX 2080 GPU.

2.1.4 Results

DARTS Cell Structure and Spatial/Temporal Functional Networks

Compared to ST-DARTS, we implemented GR-NAS to embed and then learn the best cell structure with RL and BO. We can learn the spatial/temporal functional brain networks from WM task and emotion fMRI datasets with the learned cell to learn the cell structure, GR-NAS takes approximately one GPU day. Under different settings in search algorithm parameters and different tasks, the learned cell structures might be distinct. Under the same setting, comparison between our GR-NAS model and ST-DARTS was made on WM fMRI data. As shown in Fig. 2.2, for different task designs, our GR-NAS model under RL and BO both achieved more remarkable performance than ST-DARTS, which indicates our model can learn the temporal function networks that measured by Pearson correlation coefficient (PCC) better.

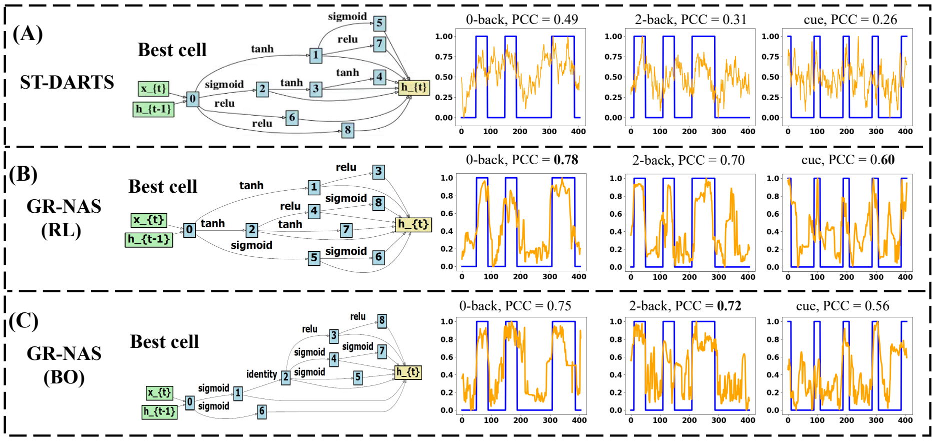


Figure 2.2: Comparison pf different cells found by ST-DARTS, GR-NAS (RF), and GR-NAS (BO). The best cells found by various models are placed in the left of each row. Furthermore, the most stimuli-correlated temporal networks learned by given cells on WM different tasks are on the right. Blue curves denote the task-design, and orange curves denote the learned temporal network dynamics.

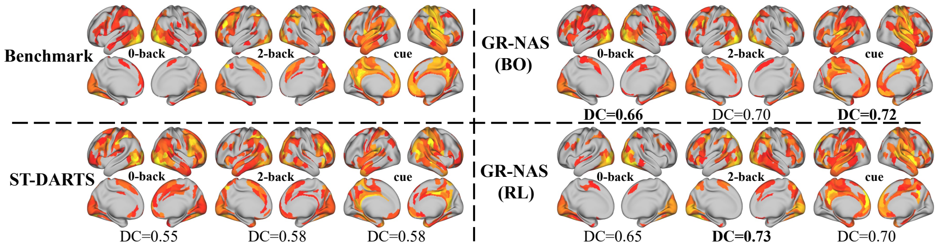


Figure 2.3: The most stimuli-correlated spatial networks learned by GR-NAS on WM fMRI tasks. The spatial benchmarks are learned by Elastic Net regression based on the temporal task-design, as in the same way we learned spatial networks. The bottom left contains the result of ST-DARTS.

We use Dice coefficient (DC) (Dice, 1945) to measure the similarity between two sets. DC is commonly used for similarity measuring between the spatial networks and the benchmark, which are derived by Elastic Net regression of fMRI data. Our first preparation is converting the functional spatial networks $Net^{(i)}$ into a Boolean type. Then, we assign one to the activated voxel, whose signal value is more than 10^{-3}, and zero to the deactivate voxel, whose signal value is less than 10^{-3}. Thus, the DC between $Net^{(i)}$ and $Net^{(j)}$ is defined as follows:

$$DC_{i,j} = \frac{2 \times |Net^{(i)} \cap Net^{(i)}|}{|Net^{(i)}| + |Net^{(j)}|} \tag{2.6}$$

Here we show the results of WM task. As shown in Fig. 2.3 , the individual best is as high as 0.73 in WM, comparing to the highest value 0.58 by ST-DARTS.

Transfer Learning on Different Task-fMRI Datasets

We also tried to use the architecture learned from the emotion fMRI data on WM data to examine the results of decomposing temporal functional brain networks.Fig. 2.4 shows the results of the spatial network learned by transfer learning compared with the ST-DARTS benchmark. Compared with the results in Fig. 2.2 and Fig. 2.4, the average correlation rates of transfer learning are slightly lower than the proposed method, but it still outperforms ST-DARTS because of an embedding process that makes the similar characteristics between the two tasks into a same target domain. This result demonstrated that our embedding method can effectively find an optimal architecture.

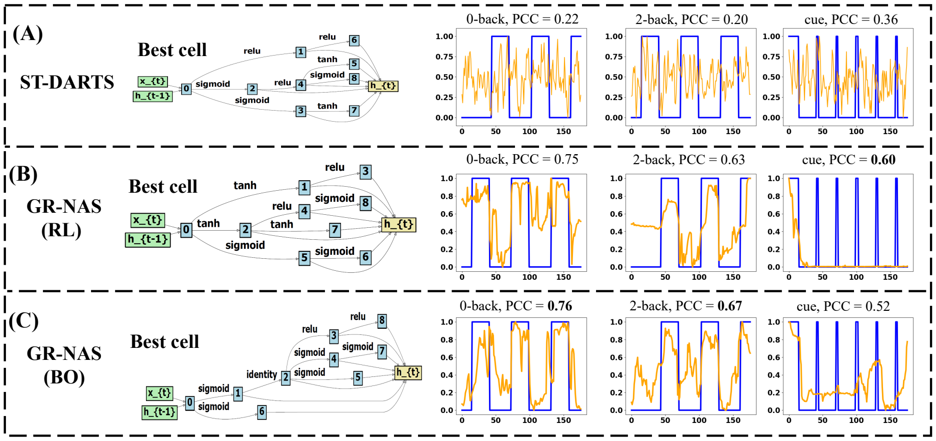


Figure 2.4: Comparison between different cells found by ST-DARTS, GR-NAS (RL) and GR-NAS (BO). The best cells found by the various models are placed in each row's left. Moreover, the most stimuli-correlated temporal networks learned by given cells on WM different tasks are on the right. Blue curves denote the task-design, and orange curves denote the learned temporal network dynamics.

Stability and Robustness of GR-NAS

To illustrate the stability and robustness of the GR-NAS model, we showed all the three runs' results of PCCs under GR-NAS between learned temporal network dynamics and task design. In Fig. 2.5, the PCCs of GR-NAS vary from 0.6 to 0.8, which is substantially higher than the previous results by ST-DARTS from 0.2 to 0.4. This result indicates that the proposed GR-NAS model can stably and robustly derive meaningful networks. The DCs of BO NAS and RF NAS are from 0.65 to 0.75 across the WM task, which outperform 0.55 to 0.65 by ST-DARTS. This result indicates that the spatial networks learned from the proposed GR-NAS model can derive similar maps with the benchmark even with different architecture search algorithms. Additionally, the variance of PCC/DC values under the proposed GR-NAS model is much less than that of PCC/DC values by ST-DARTS (Fig.2.5), which suggests the effectiveness and robustness of GR-NAS.

2.1.5 Discussion and Conclusion

This paper presented a novel GR-NAS model for optimal brain network decomposition. Unlike previous methods, we embedded the DARTS RNN cells

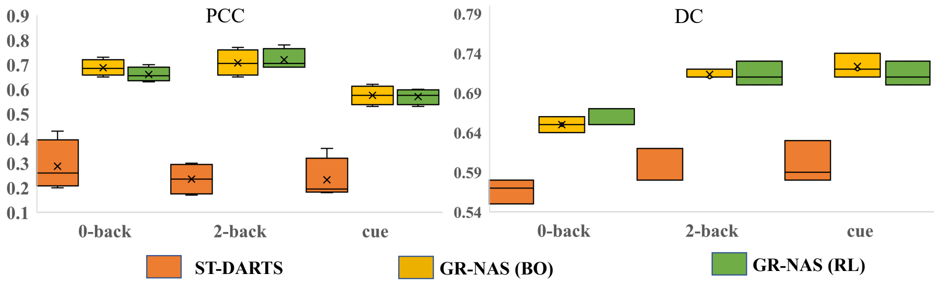


Figure 2.5: Comparison between different cells found by ST-DARTS, GR-NAS (RL) and GR-NAS (BO). The best cells found by the various models are placed in each row's left. Moreover, the most stimuli-correlated temporal networks learned by given cells on WM different tasks are on the right. Blue curves denote the task-design, and orange curves denote the learned temporal network dynamics.

into a pre-trained embedding space to preserve the topological information and learn the optimal architecture on a continuous space. Then, we implemented RL and BO in GR-NAS model to search for the optimal architectures based on the embeddings. In addition, we implemented transfer learning in this work to evaluate our GR-NAS model. It is promising that transfer learning is better than ST-DARTS, but it is slightly worse than the proposed GR-NAS model. The results indicate that our continuous embedding search space is better than a discrete search space, and it can effectively find the optimal architecture for deep neural networks.

2.2 Individual Functional Network Abnormalities Mapping via Graph Representation-based Neural Architecture Search

2.2.1 Overview

Prenatal alcohol exposure (PAE) has garnered increasing attention due to its detrimental effects on both neonates and expectant mothers. Recent research indicates that spatio-temporal functional brain networks (FBNs), derived from functional magnetic resonance imaging (fMRI), have the potential to reveal changes in PAE and Non-dysmorphic PAE (Non-Dys PAE) groups compared with healthy controls. However, current deep learning approaches for decomposing the FBNs are still limited to hand-crafted neural network architectures, which may not lead to optimal performance in identifying FBNs that better reveal differences between PAE and healthy controls. In this paper, we utilize a novel graph representation-based neural architecture search (GR-NAS) model to optimize the inner cell architecture of recurrent neural network (RNN) for decomposing the spatio-temporal FBNs and identifying the neuroimaging biomarkers of subtypes of PAE. Our optimized RNN cells with the GR-NAS model revealed that the functional activation decreased from healthy controls to Non-Dys PAE then to PAE groups. Our model provides a novel computational tool for the diagnosis of PAE, and uncovers the brain's functional mechanism in PAE.

2.2.2 Background

Prenatal alcohol exposure (PAE) can induce adverse outcomes among young mothers (Archibald et al., 2001; Jones & Smith, 1973). Though the PAE-related abnormalities include functional cognitive behavioral impairment have been reported (Bandoli et al., 2020; Mattson et al., 2019), the adverse effects on the health of young mothers are often overlooked. Based on functional magnetic resonance (fMRI), researchers have identified altered brain network organization in individuals exposed to ethanol (J. Lv, Jiang, Li, Zhu, Zhao, et al., 2015; S. Zhao et al., 2016), resulting in a significant decrease of small-worldness in spatio-temporal functional brain networks (FBNs). Notably, the spatio-temporal FBNs are fundamental components of brain activities that reflect transformations in brain function. Therefore, analyzing variations in brain function from the perspective of spatio-temporal FBNs could potentially reveal the effect across the different sub-types of PAE (J. Lv, Jiang, Li, Zhu, Chen, et al., 2015). The spatio-temporal FBNs have been extensively investigated in the neuroimaging community using deep learning approaches (H. Huang et al., 2017; Q. Li et al., 2019; Y. Zhao et al., 2018). For example, a deep sparse recurrent autoencoder (DSRAE) (Q. Li et al., 2019) was developed to simultaneously decompose FBNs and demonstrated the effectiveness of recurrent neural network (RNN) models in extracting neuroscientifically meaningful spatio-temporal networks from 4D fMRI data. Generally, designing appropriate or optimal neural network architectures manually as aforementioned approaches is a challenging and time-consuming process that heavily relies on domain knowledge and experience. To address these challenges, neural architecture search (NAS) related approaches have been proposed for identifying the optimal architectures for FBNs analysis [10, 11]. Among them, differentiable neural architecture search (DARTS) (H. Liu et al., 2018) has improved the efficiency of searching optimal RNN architectures while maintaining comparable performances, which has been adopted in spatio-temporal FBNs decomposition, called spatio-temporal DARTS (ST-DARTS) (Q. Li et al., 2021). Despite the efficiency of the DARTS framework that relaxed the operations on inner nodes with the maximum approximation for the optimization process, DARTS-based algorithms are still limited by the discrete space among the inner nodes in RNN cells. Additionally, these algorithms do not consider the topological information among inner nodes within RNN cells (H. Liu et al., 2018), and may be trapped in local optimum that further degrade performance. Considering that PAE-related FBNs are suggested to be associated with functional connectivity among brain regions, which is a kind of topology of the human brain (Wozniak et al., 2017), applying the DARTS-based methods directly on assessing the brain functions with PAE may be negatively affected due to the lack of topological information. In this work, we use a novel graph representation-based neural architecture search (GR-NAS) to optimize the RNN cell architecture for decomposing the spatio-temporal FBNs (Dai et al., 2022) and identifying the neuroimaging biomarkers of subtypes of PAE. Specifically, to optimize the DARTS's discrete searching process, we represent the RNN cell architectures as graphs and embed them into a latent continuous search space via graph isomorphism network (GIN) (K. Xu et al., 2018) encoder that is the graph representation process. Then, the GR-NAS can utilize the embedded graph to search the optimal RNN cells in a continuous space and preserve the topological information. In this paper, we employed reinforce learning (RL) as the search strategy engine for optimizing RNN cells on the PAE task fMRI dataset (Santhanam et al., 2009), which includes the normal controls, exposed Non-dysmorphic PAE (Non-Dys PAE) and exposed dysmorphic PAE participants. The results demonstrate the robustness and the reliability of the identified biomarkers of PAE in both group-wise and individual manner. To our best knowledge, this paper is one of the earliest contributions to PAE FBN analysis with NAS-based deep models, providing a new perspective for the abnormal brain early diagnosis with fMRI data.

2.2.3 Materials and Method

Overview

Fig. 2.6 presents the overview of the GR-NAS model. The general purpose of this work is to learn the optimal RNN cells for decomposing the spatio-temporal FBNs from fMRI data for the subtypes of PAE. RNN cells' graph-structured architectures are embedded with the GIN encoder to learn the pre-

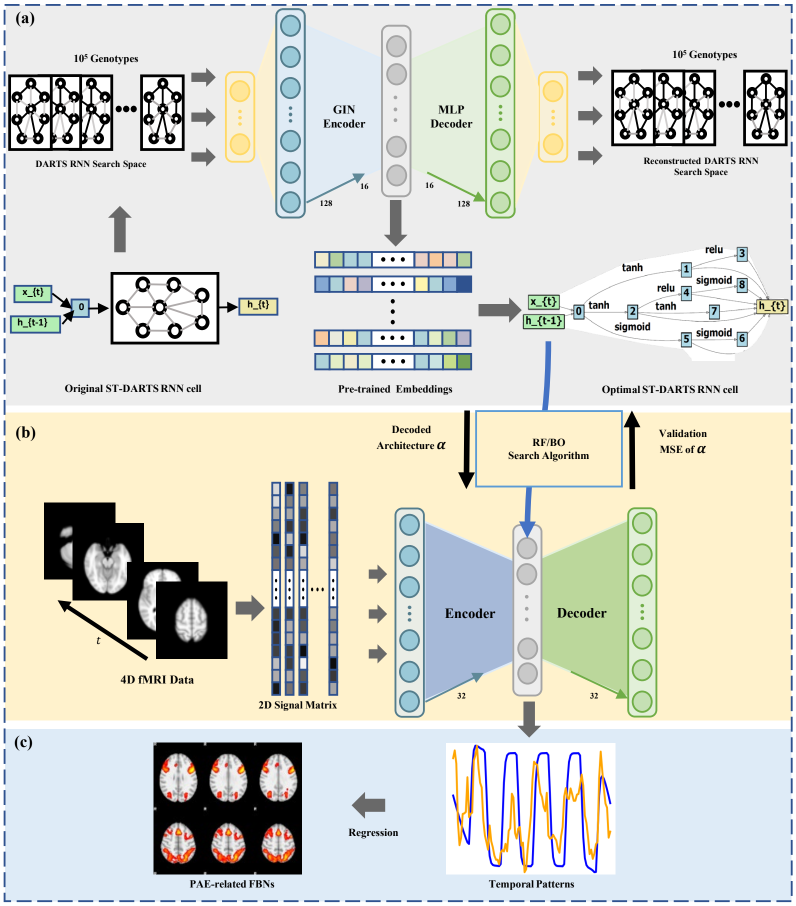


Figure 2.6: Our framework of GR-NAS. (a) A GIN encoder and a multiple layer perceptron (MLP) decoder are used to embed the RNN cell architectures to obtain pre-trained embeddings. (b) RL search strategy is used on the pre-trained embeddings to search for the optimal architecture based on the architecture performance estimation. (c) Decomposed spatio-temporal functional network learning with the learned optimal cell architecture.

trained embeddings. Then, RL search strategy is used to find the optimal architecture of the SOTA ST-DARTS, which is taken as the baseline in this paper. Finally, we utilize the acquired RNN cells to analyze the 4D fMRI data, highlighting the distinctions from the original RNN cells and obtaining a high-level feature map in the latent layer for decomposing spatio-temporal FBNs.

Data Description and Pre-processing

In this paper, we adopted 44 participants' fMRI data that were scanned at the Biomedical Imaging Technology Center of Emory University. The 44 participants were from 3 groups, which were the exposure with presence of dysmorphic signs group (PAE, 14 participants), the exposure with the absence of dysmorphic signs group (Non-Dys PAE, 14 participants) and the unexposed normal controls group (Control, 16 participants) (Santhanam et al., 2009). All the participants were with an age range of 20 to 26. Ten task blocks of subtraction arithmetic and letter-matching control stimuli were alternated during the experiment, and in total 100 time points were used. Important pre-processing steps include motion correction, slice time correction, spatial smoothing, and global drift removal. FSL-FLIRT was used to register the pre-processed volumes against the Montreal Neurological Institute (MNI) template. In order to focus on the fluctuations of fMRI signals, we normalized each extracted signal with mean of 0 and standard deviation of 1.

Spatio-temporal Differentiable Architecture Search (ST-DARTS)

The ST-DARTS is the SOTA DARTS-based RNN cell optimization algorithm in the field of neuroimaging, which is taken as the baseline in this paper. The ST-DARTS RNN cell is based on the vanilla RNN cell [13, 18], which is defined as:

$$h_t = tanh(W_{xh}X_t + U_{hh}h_{t-1} + b_h) \tag{2.7}$$

in which, the h_t is the RNN cell's hidden state that maintains the sequence memory of the temporal information of brain dynamics, x_t is the input of the fMRI signal matrix,W_{xh} and U_{hh} are the weights of the current input and the previous hidden state, respectively,b_h is the bias, and $tanh(\cdot)$ is the activation function in RNN cell that implements the non-linearity to squash the activations to the range [-1,1]. Then the RNN cell's output y_t could be defined based on the hidden state h_t and the weights of output V_{yh}:

$$y_t = V_{yh}h_t \tag{2.8}$$

Inherited from the vanilla RNN cell, for each ST-DARTS RNN cell, there are two inputs and a single output, which are the current step input x_t (the volume sample on the t-th time point), the hidden state from the previous step h_{t-1}, and the concatenation of all the intermediate nodes y_t. Each ST-DARTS RNN cell is a directed acyclic graph consisting of an ordered sequence of N nodes. The candidate operation choices on nodes are relaxed by softmax to make the search space continuous as follows:

$$\bar{o}^{i,j}(node) = \textstyle\sum_{o \in O} \frac{exp(\alpha_o^{(i,j)})}{\sum_{o'} exp(\alpha_{o'}^{(i,j)})} o(node) \tag{2.9}$$

where the function $o(\cdot)$ indicates the operation that is applied on the inner node $node^{(i)}$ of the cell. For more specifically, $\bar{o}^{(i,j)}$ represents the mixed operation from $node^{(i)}$ to $node^{(j)}$, and o' denotes the one-step forward model's operation. And node denotes the collect of the inner nodes of such ST-DARTS RNN cell. $node^{(i)}$ represents the i^{th} node in the cell architecture that is a latent representation. $\alpha_o^{(i,j)}$ is the operation mixing weight of the given operation $o(\cdot)$ from $node^{(i)}$ to $node^{(j)}$. After the jointly learning process of the cell architecture parameter α and the entire ST-DARTS RNN architecture weights w, the discrete architecture can be obtained by replacing the mixed operation $\bar{o}^{(i,j)}$ with the most likely operation. Though the ST-DARTS RNN cell is searched after relaxing the discrete operations into a continuous space, such cell only focuses on the operations between the inner nodes and ignores the topological information within RNN cells. In other words, the current ST-DARTS RNN cell makes the topological information actually in the discrete space, instead of in the continuous space, which would be improved with a future NAS method to convert the whole acyclic graph into a continuous space to promote spatio-temporal FBN decomposition.

Graph Representation Neural Architecture Search

In this work, we use the novel GR-NAS model to search for the optimal RNN cell architecture, and then apply it to the PAE-related FBNs decomposition. GR-NAS employs a variational graph isomorphism autoencoder to embed topological graph of RNN cell into a continuous searching space; then RL strategy is used to explore the optimal architecture in this searching space.

Variational Graph Isomorphism Autoencoder: During the embedding process, each RNN cell was represented as a directed acyclic graph (DAG). We denote the DAG as $DAG = (V, E)$, where V is the set of nodes and E is the set of edges. Each node of the DAG is associated with one of four predefined activation operations, including tanh, identity (indicating there is a connection between two nodes without activation operation), sigmoid and ReLU. Therefore, the DAG can be represented by two matrices: one upper triangular adjacency matrix $A \in \mathbb{R}^{N \times N}$ that encodes the connections among N nodes and one-hot operation matrix $X \in \mathbb{R}^{N \times K}$ that records the operation of each node. To preserve the topological information of RNN cell, GIN was used to encode the graph-structured architectures into embedding space $Z \in \mathbb{R}^{N \times K}$ as follows:

$$q(Z|X,A) = \prod_{i=1}^{n} q(z_i|X,A), with\ q(z_i|X,A) \sim \mathcal{N}(z_i|\mu_i, diag(\sigma^2)) \tag{2.10}$$

where a one-layer GIN encoder $q(\cdot)$ is used to embed the adjacency matrix A and operation matrix X into the embedding vector Z. z_i represents the i^{th} value of the latency embedding vector Z. μ,σ are the mean and variance of approximation $q\left(z_i|X,A\right)$, respectively. $\mathcal{N}(\cdot)$ indicates Gaussian distribution. Then the embedding matrix H was as:

$$H = MLP((1+\epsilon)\cdot X + A \times X) \tag{2.11}$$

where H is the output node embedding matrix, ϵ is a trainable bias, MLP denotes a multi-layer perceptron, in which each layer is a linear-batchnorm-ReLU triplet. With Eq. 2.11, based on H, the mean μ and the variance σ could be obtained from H as the Gaussian distribution parameters to approximate $q\left(Z \mid X, A\right)$. We use generative model $p(\cdot)$ to obtain the reconstructed connection $\hat{A}$ and one-hot operation matrix $\hat{X}$ from the latent variable Z. The one-layer MLP decoder is:

$$P(\widehat{A}|Z) = \textstyle\prod_{i=1}^{N}\prod_{j=1}^{N} p(\widehat{A}_{ij}|z_i,z_j), with\ P(\widehat{A}_{ij}=1|z_i,z_j) = \vartheta(z_i^T z_j) \tag{2.12}$$

$$P(\widehat{X} = [k_1,\ldots,k_n]^T|Z) = \textstyle\prod_{i=1}^{N} P(\widehat{X} = k_i|z_i) = \prod_{i=1}^{N} softmax(WZ+b) \tag{2.13}$$

where ϑ is the logistic sigmoid function and $\hat{A}_{ij}$ indicates the element of $\hat{A}$. k_i indicates the i^{th} operation. We optimize the GIN autoencoder by maximizing the lower bound $\mathfrak{L}$ of variational parameters as:

$$\mathfrak{L} = \mathbb{E}_{q(Z|X,A)}[logp(\hat{X},\hat{A}|Z)] - KL[q(Z|X,A)||p(Z), \tag{2.14}$$

where we assume the adjacent matrix A and the operation matrix X are conditionally independent here. The $\mathbb{E}$ term indicates the expectation, and the KL term measures the differences between the posterior distribution $q(\cdot)$ and the prior distribution $p(\cdot)$. Then, the full-batch gradient descent was performed and the parameterization scheme was used to generate random noises during the training process as the regularization (Zou & Hastie, 2005).

Search Strategy on Embeddings: With the obtained RNN-based embeddings cell on the PAE-related fMRI data, we then employed down-stream search methods to evaluate our model on pre-trained embeddings. During the GR-NAS-RL process, the state is the 16-dimension embedding. The action is the movement on one of the embedding's 16 dimensions. The reward is the reconstruction loss from current state. Pre-trained embedding is passed to the ST-DARTS RNN cell to evaluate current state and then obtain the next action and state based on the L^2 distance to minimize the reward. In order to prove the robustness and stability of our framework, we also took use of Bayesian optimization (BO) as an alternative search strategy for comparison. During the GR-NAS-BO process, the deep networks for global optimization was used to search for the optimal architecture on the 16-dimensional embeddings. One-layer adaptive basis regression network is employed for modeling the distribution over functions with 128 hidden dimensions. We use Adam optimizer here and set the learning rate to be 0.01. After the searching process, the derived best ST-DARTS RNN cell architecture will be fed into the GR-NAS model for future spatio-temporal function network learning for PAE.

PAE Spatio-temporal Functional Network Learning

After GR-NAS searching on the embeddings, optimal ST-DARTS RNN cells are achieved for PAE subtypes' spatio-temporal functional network. With the latent ST-DARTS RNN cells, the temporal network dynamics were achieved, and the spatial networks were derived with the Elastic Net regression further (Y. Zhao et al., 2018; Zou & Hastie, 2005). We used the Pearson's correlation coefficient (PCC) to evaluate how consistent temporal networks are with the true brain states that are stimulated by the task design. For each extracted spatial network, we use the Dice coefficient (DC) (Dice, 1945) to measure the similarity between two networks (the derived FBNs $Net^{(i)}$ and the benchmark networks). More specifically, $Net^{(0)}$ is the benchmark derived from the true brain state series that stimulated by tasks with Elastic Net regression, and $Net^{(1)}$ to $Net^{(32)}$ are the brain spatial networks that are derived PAE-related FBNs. The Elastic Net regression is an effective way to regress the temporal series to the spatial features that could take advantage of both Lasso and Ridge regressions (Y. Zhao et al., 2018; Zou & Hastie, 2005). In the temporal functional network learning process, the epoch was set as 200 to guarantee convergence, the batch size was set as 128. Each cell has 8 nodes that are set in the same ways as in [12, 13, 15].The hidden layer size was set as 32, and the initial learning rate was set as 20. We implemented the proposed GR-NAS model with PyTorch 1.4.1 on a single RTX 2080 GPU.

2.2.4 Results

The framework has been applied to the dataset of three groups of PAE related participants: Control, Non-Dys PAE and PAE. The severity of PAE is in the order of Control < Non-Dys PAE < PAE. The common networks are learned for all three group and the group-wise statistic is applied to each group separately.

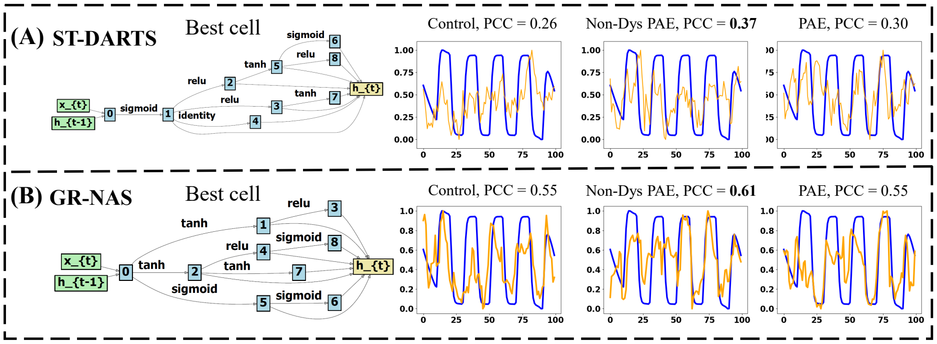


Figure 2.7: Comparisons across different cell architectures derived from ST-DARTS and GR-NAS. The best cell architectures learned by the search models are shown on the left of each row. And the most task stimuli-correlated temporal networks learned by such cell architectures are shown on the right. The blue curves denote the task-design convolved with HRF and the orange curves denote the learned temporal network dynamics.

Optimized ST-DARTS Cell Architecture

We implemented GR-NAS to embed and then learn the best cell architecture with the RL search strategy. With the learned cell architecture, we can produce the spatio-temporal functional brain networks from PAE fMRI datasets. In order to get the cell architecture, GR-NAS takes approximately one GPU day. Under the same super-parameter setting, our GR-NAS model was compared with ST-DARTS based on the PAE-related task fMRI data. As shown in Fig. 2.7, for different groups, our GR-NAS model could achieve greater performance than ST-DARTS, which means our model can learn the temporal functional networks better. The temporal results with the searched cell genotypes are shown in Fig. 2.7. The task design curve convolved with hemodynamic response function (HRF) is visualized as blue curves, which is used for calculating the PCCs with the learned temporal net-works from GR-NAS model. For ST-DARTS, the PCC is around 0.3 and the best PCC is 0.37 for the Non-Dys PAE group. The PCCs for all the groups under GR-NAS are all higher than 0.55 and the best PCC is 0.61 for the Non-Dys PAE group. Apparently, the GR-NAS model performs much better than the SOTA original ST-DARTS.

Spatio-temporal Functional Networks of Subtypes of PAE

In order to illustrate the brain spatial networks, we show the most typically derived group spatial functional brain networks in Fig. 2.8. To avoid the effect of the noises, we set the z-score maps with a threshold >1.65. As reported in (J. Lv, Jiang, Li, Zhu, Zhao, et al., 2015; Santhanam et al., 2009), the activation regions tend to shrink by the increment of severity of PAE effect, which means the number of activated voxels would decrease from Control group to the severe PAE group.

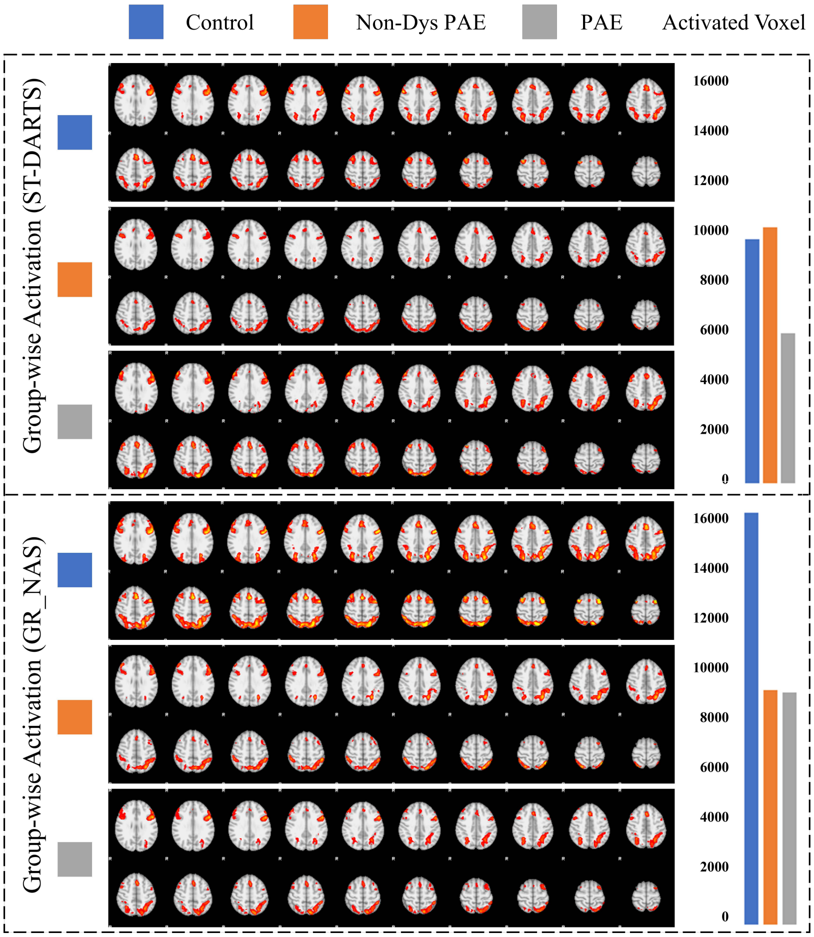


Figure 2.8: Spatial network of group-wise activation. With GR-NAS, the voxel number (V) of the group networks decreases across three groups, i.e., V(Control) > V(Non-Dys PAE) > V(PAE). The top graph shows the activation of spatial map for each group.

As shown in Fig. 2.8, based on the activation patterns and the quantitative voxel numbers of the activation region across three groups under three different methods share the same pattern: Control >Non-Dys PAE > PAE, which is consistent with the characteristic of PAE (J. Lv, Jiang, Li, Zhu, Zhao, et al., 2015). More specifically, the temporal and parietal brain regions are activated clearly with all the three models, and the area of activation has been decreased from Control to the PAE patients. This is consistent with the previous litera-ture, which has already shown that such temporal and parietal brain regions' activation decrease is related to the mental disease (Calhoun et al., 2009). For the ST-DARTS model, the number of activated voxels does not decrease between the Control group and the Non-Dys PAE group. However, according to the visualization of the brain patterns, the key regions are cut down. With GR-NAS, the pattern is shown more clearly, which proves that GR-NAS could obtain more variable and neuroscientific brain networks.

Specific Network Analysis

In order to support the sub-networks identified by the group-wise analysis, we selected the top three temporal correlated and anti-correlated networks of the Control group that exhibited high correlation with the task design HRF and the top three net-works that exhibited high anti-correlation with the task design HRF. For each selected top temporal network in the Control group, we selected the most DC-correlated net-works in the Non-Dys PAE group and the PAE group based on the DC. As shown in Table 2.1, most of the DCs between the Non-Dys PAE group and the Control group are greater than those between the PAE group and the Control group. With GR-NAS, the highest DC between Control and Non-Dys PAE groups is 0.7, and a 0.02 decrease is occurred between the Control and PAE group. This also proves that the disease severity of the Non-Dys PAE group is not as severe as that of the PAE group, which produces a straightforward evidence to reveal the PAE mechanism.

Individual-wise Brain Spatial Networks Analysis

Furthermore, in order to prove our findings are robust and stable, we also show the individual-wise brain spatial networks with both GR-NAS-RL and GR-NAS-BO in Fig. 2.9. Similar to the group-wise results, the activated brain voxels decrease sharply from the Control group to the Non-Dys PAE group then to the PAE group. Though the activations on the individual-wise brain are affected by the noise and lead to the uneven cluster, the activated brain voxels decreasing tendency is clearly same as the group-wise results. Especially with the GR-NAS-RL method, for both Network #9 and Network #32, the tendency of the decrease is clearer. The activated regions are clustered around the parietal and temporal areas, which are the key areas for cognitive conception (Barch

Table 2.1: Networks in the Control group, the matched networks in the Non-Dys PAE group and the PAE group selected by DC.

	GR-NAS (RL)		
	Control	Non-Dys PAE	PAE
The top temporal anti-correlated networks	#25	#28 (0.63)	#6 (0.63)
	#31	#3 (0.7)	#19 (0.68)
	#29	#3 (0.63)	#12 (0.61)
The top temporal correlated networks	#4	#28 (0.61)	#15 (0.61)
	#12	#4 (0.63)	#12 (0.64)
	#7	#8 (0.62)	#31 (0.62)

et al., 2013). Quantitively, for the Network #9 with GR-NAS-RL method, the number of activated voxels decreases from 54101 to 30202 then to 27385, and for the Network #32, the activated voxels goes down from 60667 to 32776 and then to 10682. This is similar with the tendency that the activation regions would shrink with the increment of severity of PAE effects [23]. On the other hand, with the GR-NAS-BO method, the number of activated voxels of Network #17 decreases from 55219 to 34860 and then to 34354, and from 68236 to 35966 and then to 34689 for Network #16. Based on Fig. 2.9, the difference between Non-Dys PAE and PAE group is consistently less than the difference between the Control group and the Non-Dys PAE group, no matter each kind of search strategy, which may provide evidence for the diagnosis of Non-Dys PAE.

2.2.5 Discussions and Conclusions

In this paper, we utilized a novel GR-NAS model for optimal brain network decomposition. Unlike previous methods, we embedded the RNN cells into a pre-trained embedding space to preserve the topological information so we can learn optimal architecture in a continuous space. Then, we implemented alternative searching strategies to search for the optimal architectures on the embeddings. Our approaches have been applied to three groups of participants affected by PAE to different degrees, namely, the Control group, the Non-Dys PAE group and the PAE group. The experimental results have suggested that our method can detect the temporal and parietal networks across three groups, while such networks are affected by an increment of PAE severity (i.e., the activated regions shrink according to the PAE degree).

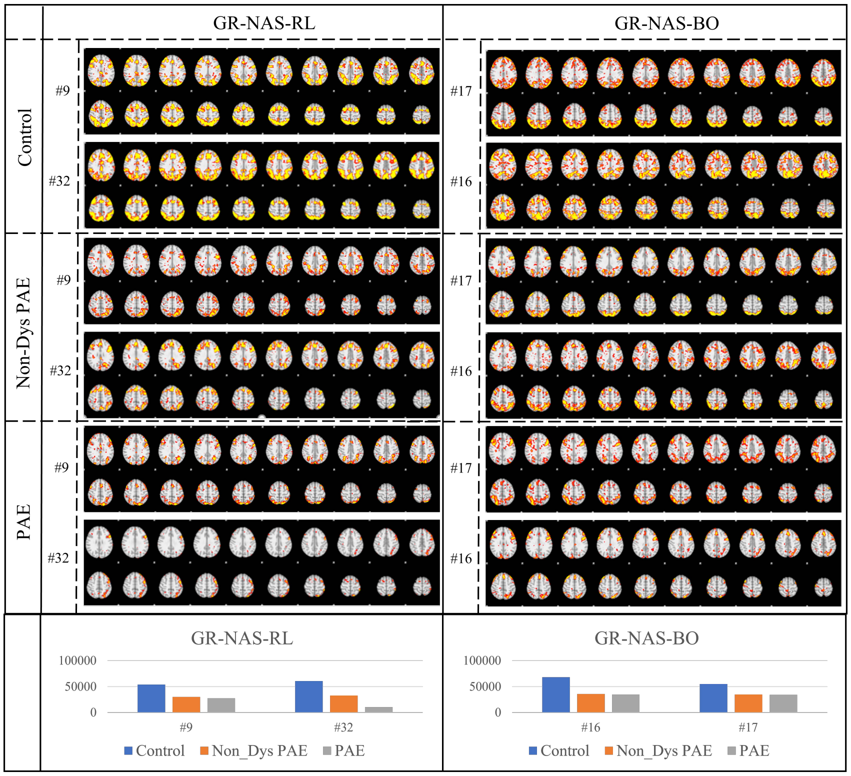


Figure 2.9: Individual-wise brain spatial networks. The index under the graph means the i^{th} network in that group. The bottom picture shows the activated voxel for each brain network.

Chapter 3

Hierarchical Semantic Tree Concept Whitening for Interpretable Image Classification

3.1 Overview

With the popularity of deep neural networks (DNNs), model interpretability is becoming a critical concern. Many approaches have been developed to tackle the problem through post-hoc analysis, such as explaining how predictions are made or understanding the meaning of neurons in middle layers. Nevertheless, these methods can only discover the patterns or rules that naturally exist in models. In this work, rather than relying on post-hoc schemes, we proactively instill knowledge to alter the representation of human-understandable concepts in hidden layers. Specifically, we use a hierarchical tree of semantic concepts to store the knowledge, which is leveraged to regularize the representations of image data instances while training deep models. The axes of the latent space are aligned with the semantic concepts, where the hierarchical relations between concepts are also preserved. Experiments on real-world image datasets show that our method improves model interpretability, showing better disentanglement of semantic concepts, without negatively affecting model classification performance.

3.2 Background

Machine learning interpretability has recently received considerable attention in various domains (De Clercq et al., 2018; Du et al., 2019; Koh et al., 2020; Murdoch et al., 2019). An important challenge that arises with deep neural networks (DNNs) is the opacity of semantic meanings of data representations in hidden layers. Several types of methods have been proposed to tackle the problem. First, recent works have shown that some neurons could be aligned with certain high-level semantic patterns in data (Olah et al., 2017; B. Zhou et al., 2018). Second, it is possible to extract concept vectors (Kim et al., 2018) or clusters (Ghorbani, Wexler, et al., 2019) to identify semantic meanings from latent representations. However, these methods are built upon the assumption that semantic patterns are already learned by DNNs, and the models would admit the post-hoc method of a specific form. There is no guarantee that the assumption holds true for any model, especially when meaningful patterns or rules may not be manifested in the model, thus leading to over-interpretation (Murdoch et al., 2019; Rudin, 2019a). Meanwhile, although many post-hoc explanation methods are proposed with the expectation of improving or debugging models, it is challenging to achieve this goal in practice. Although we could collect human annotations to guide prediction explanations and improve model credibility (Chang et al., 2021; J. Wang et al., 2018), manually labeling or checking semantic concepts is rather difficult. Unlike explaining individual predictions, which is a local and instance-level task, extracting concepts provides a global understanding of models, where manual inspection of such interpretation is time-consuming and much harder, if not impossible.

Instead of relying on post-hoc approaches, we aim to instill interpretability as a constraint into model establishment. For example, explanation regularization is proposed in (Ross & Doshi-Velez, 2018), but it constrains gradient magnitude instead of focusing on semantic concepts. Meanwhile, β-VAE and its variants (R. T. Chen et al., 2019; Higgins et al., 2017) add independence constraints to learn disentangled factors in latent representations, but it is difficult to explicitly specify and align latent dimensions with semantic meanings. Ideally, we want to construct DNNs whose latent space could tell us how it is encoding concepts. The recent decorrelated batch normalization (DBN) method (L. Huang et al., 2018a) normalizes representations, providing an end-to-end technique for manipulating representations, but it is not directly related to interpretability.

In this work, we propose a novel Hierarchical Semantic Tree Concept Whitening (HaST-CW) model to decorrelate the latent representations in image classification for disentangling concepts with hierarchical relations. The idea of our work is illustrated in Fig. 3.1. Specifically, we define each concept as one class of objects, where the concepts are of different granularities and form a hierarchical tree structure. We decorrelate the activations of neural network layers, so that each concept is aligned with one or several latent dimensions. Different from the traditional DBN method (Fig. 3.1a) that only treats different concepts as being independent, our method could leverage the underlying hierarchically related organization of label concepts specified by the domain knowledge (Fig. 3.1b). The consideration of relations between different concepts is crucial in many real-world applications. For example, in the healthcare domain, the relationship of different disease stages (concepts) may reflect the progression of the disease, which is significant for reversing pathology (L. Wang et al., 2020; L. Zhang, Wang, et al., 2020, 2021). Also, in the precision agriculture domain, real-time monitoring of interactions of multiple agricultural objects (concepts) with each other and with the environment are crucial in maintaining agro-ecological balance (De Clercq et al., 2018). In our model, a novel semantic constraint (SC) loss function is designed to regularize representations. As a result, the data representations of two concepts with higher semantic similarity will be closer with each other in the latent space. Moreover, a new hierarchical concept whitening (HCW) method is proposed to decorrelate different label concepts hierarchically. We evaluated the proposed HaST-CW model using a novel agriculture image dataset called Agri-ImageNet. The results suggest that our model could preserve the semantic relationship between the label concepts, and provide a clear understanding of how the network gradually learns the concept in different layers, without hurting classification performance.

3.3 Related Work

Post-Hoc Interpretation. Post-Hoc interpretation can be divided into approaches that explain predictions or models (Du et al., 2019; Murdoch et al., 2019). Prediction-oriented interpretation aims to develop faithful and robust measures to quantify feature importance towards individual predictions for identifying those features (e.g., pixels, super-pixels, words) that made most contributions (Bach et al., 2015; Ghorbani, Abid, et al., 2019; Lundberg & Lee, 2017; Ribeiro et al., 2016; Selvaraju et al., 2017; Smilkov et al., 2017). Model-oriented interpretation analyzes behaviors of neural networks either by characterizing the function of model components (Olah et al., 2017; Simonyan et al., 2013; Zeiler & Fergus, 2014) or analyzing semantic concepts from latent representations (Bau et al., 2018; Ghorbani, Wexler, et al., 2019; Kim et al., 2018; Mu & Andreas, 2020). The proposed method also targets concept-level interpretation

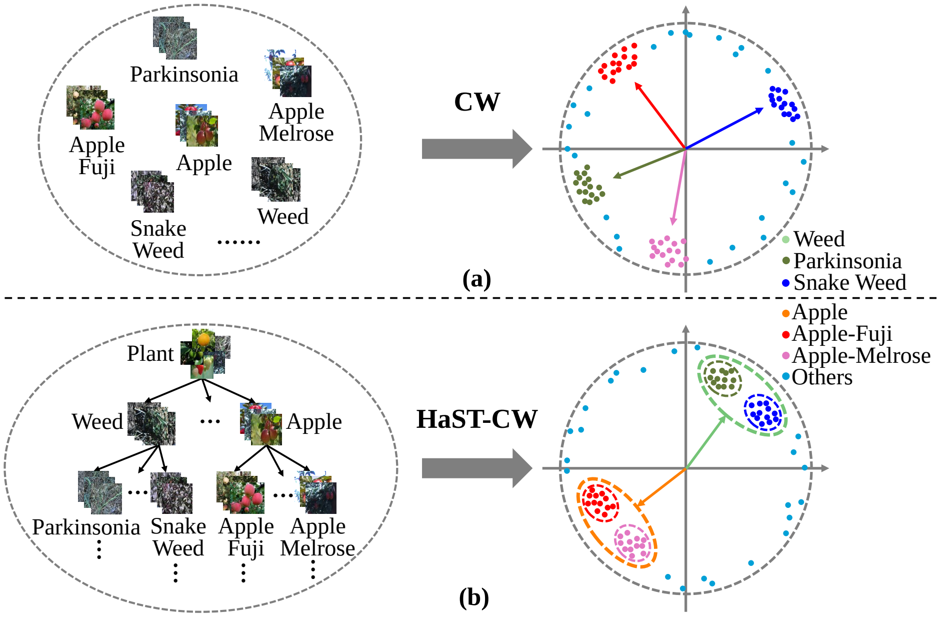


Figure 3.1: The intuition behind HaST-CW. (a) Distribution of discrete concepts in the latent space after applying concept whitening. (b) Distribution of hierarchical concepts after applying HaST-CW.

in deep neural networks. Different from post-hoc techniques that focus on discovering existing patterns in models, the newly proposed HaST-CW proactively injects concept-related knowledge into training and disentangles different concepts to promote model interpretability.

Inherently Interpretable Models. Another school of thought favors building inherently explainable machine learning models (Z. Chen et al., 2020; Rudin, 2019b). Some approaches design models that highlight prototypical features of samples as interpretation. For example, Chen et al. (C. Chen et al., 2018) classifies images by dissecting images into parts and comparing these components to similar prototypes towards prediction. Li et al. (O. Li et al., 2018) designs an encoder-decoder framework to allow comparisons between inputs and the learned prototypes in latent space. Some other works such as β-VAE and its variants (R. T. Chen et al., 2019; Higgins et al., 2017) regularize representation learning for autoencoders to produce disentangled factors in representation dimensions, but the semantic meaning of each dimension remains unknown without further manual inspection. In contrast, our method attempts to explicitly align latent dimensions with specific semantic concepts contained in external knowledge. A recent technique called Concept Whitening (CW) (Z. Chen et al., 2020) constrains the latent space, after revising Batch Whitening (L. Huang et al., 2018b; Siarohin et al., 2018), such that it aligns with predefined classes. Our method attempts to infuse more complex knowledge of concept relations into representation learning.

Applying Whitening to Computer Vision. Whitening is a standard image preprocessing technique, which refers to transforming the covariance matrix of input vectors into the identity matrix. In fact, the well-known Batch Normalization (Ioffe & Szegedy, 2015) can be regarded as a variant of whitening where only the normalization process is retained. There are many works in deep learning that describe the effectiveness of whitening (Cogswell et al., 2015; Luo, 2017; Pal & Sudeep, 2016) and the process of finding the whitening matrix (Desjardins et al., 2015). Our work further takes semantics into consideration during the whitening process towards more interpretable representation learning.

3.4 Methodology

3.4.1 Overview

The proposed HaST-CW model aims to preserve the underlying hierarchical relationship of label concepts, as well as to disentangle these concepts by decorrelating their latent representations. To achieve this goal, we leverage the hierarchical tree structure of the label concepts extracted from specific domain knowledge (Sec. 3.4.2). Then, the obtained structure of label concepts is used as prior knowledge to be instilled into the model for guiding the representation learning process. There are two key components in the knowledge instillation process – the hierarchical concept whitening (HCW) module and the semantic constraint (SC) loss, which will be elaborated in Sec. 3.4.3 and Sec. 3.4.4, respectively.

3.4.2 The Hierarchical Semantic Tree of Concepts

In this work, we used a newly collected and curated Agri-ImageNet dataset to develop and evaluate the HaST-CW model. There are 9173 high quality images in Agri-ImageNet, covering 21 different types of agricultural objects. Taking each type of agricultural object as one class, we have 21 label concepts in total. Some pairs of agriculture objects have the supertype-subtype relationship between them, so we obtain the parent-child relationship between the corresponding labels. As a result, a tree structure is built to represent the underlying

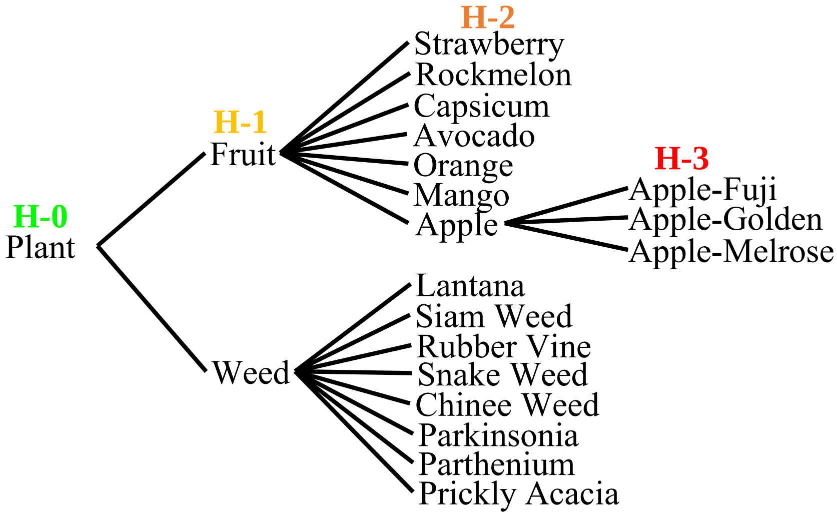


Figure 3.2: Hierarchical Tree Structure of Concepts.

hierarchically related organization of label concepts, which is shown in Fig. 3.2. Two concepts connected in the tree structure means they have parent-child relationship, where the parent is located at the lower hierarchy level. Besides the parent-child relation, we further introduce two notions – brother and cousin. If two concepts have the same parent, then they are brothers. If the parents of two concepts are brothers, then the two concepts are cousins. According to the laws of inheritance: (1) objects with the parent-child relation should be more similar than those with the uncle-child relation (vertical parent-child relationship); and (2) the traits of brothers should be more similar than cousins (horizontal brother-cousin relationship). An effective model should be able to capture both of the vertical relationship and horizontal relationship, so that the representation of any concept in the latent space should be closer to its parent than uncles, and closer to brothers than cousins. For our HaST-CW model shown in Fig. 3.3, a new HCW module (Sec. 3.4.3) is proposed to preserve the vertical relationship, and a novel SC loss (Sec. 3.4.4) is proposed to preserve the horizontal relationship.

3.4.3 Hierarchical Concept Whitening

The hierarchical concept whitening (HCW) module is one of the key components in the HaST-CW model, which aims to disentangle different label con-

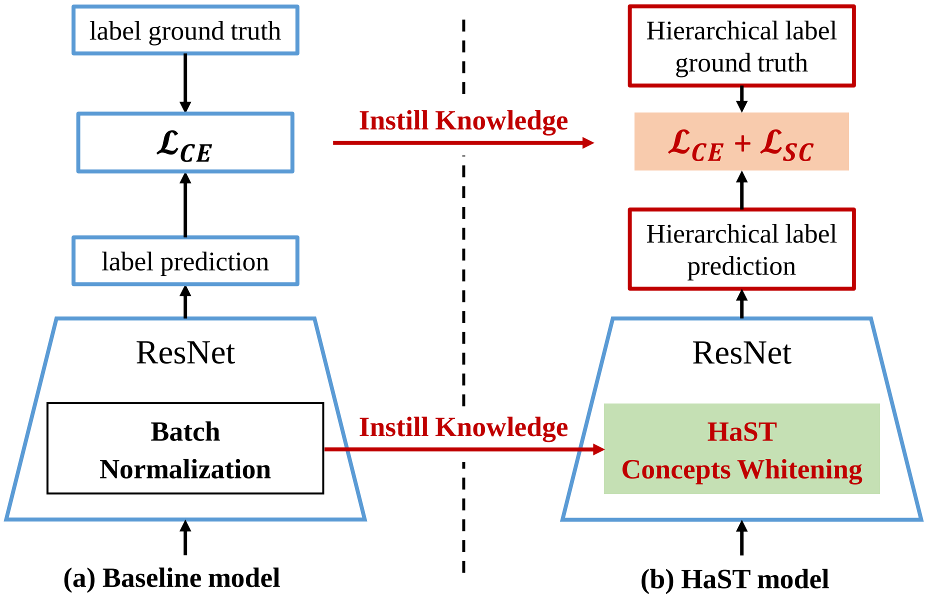


Figure 3.3: The architecture of HaST-CW model.

cepts while preserving their underlying hierarchical relationship. Specifically, in this work, the set of label concepts were denoted by $C = \{C_i\}_{i=1}^{N_c}$, where C_i represents the i^{th} concept and $N_c = 21$ is the number of concepts. For C_i, its parent, children, brothers and cousins were denoted as $C_{i.\mathcal{P}}$, $\{C_{i.children}\}$, $\{C_{i.\mathcal{B}}\}$ and $\{C_{i.\mathcal{C}}\}$, respectively. A dataset is denoted as $\mathcal{D}\{\mathbf{x}_i, y_i\}_{i=1}^{n}$. We use $\mathbf{X}^{C_i} = \{\mathbf{x}_j^{C_i}\}_{j=1}^{n_i}$ to denote the set of i^{th}-class samples labeled by C_i.

In traditional whitening transformation (Z. Chen et al., 2020), during the training process, data samples are first fed into the model in mini-batches to obtain the latent representation matrix $\mathbf{Z}_{d\times n}$, where n is the mini-batch size and d is the dimension of latent representation. We use ResNet as the model backbone in this work. Then a transformation ψ is applied to decorrelate and standardize $\mathbf{Z}_{d\times n}$:

$$\psi(\mathbf{Z}) = \mathbf{W}(\mathbf{Z} - \mu\mathbf{1}_{n\times 1}{}^T), \tag{3.1}$$

where $\mathbf{W}_{d\times d}$ is the orthogonal whitening matrix, and $\mu = \frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i$ is the sample mean. A property of representation whitening is that $\mathbf{Q}^{\mathbf{T}}\mathbf{W}$ is still a valid whitening matrix if $\mathbf{Q}$ is an orthogonal matrix. We leverage this property for interpretable representation learning. In our model, besides decorrelation and standardization, we expect that the transformed representation of samples from concept C_i, namely $\mathbf{Q}^{\mathbf{T}}\psi(\mathbf{Z}^{C_i})$, can align well with the i^{th} axis of latent space. Meanwhile, the underlying hierarchical relationship of concepts should also be preserved in their latent representations. That is, we need to find an orthogonal matrix $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{N_c}]$ with two requirements: (1) $\mathbf{Z}^{C_i}$ should be most activated by $\mathbf{q}_i$, i.e., the i^{th} column of $\mathbf{Q}$; (2) $\mathbf{Z}^{C_i}$ should also be activated by $\{\mathbf{q}_c\}$, where $c \in \{C_{i.children}\}$ is the child of concept C_i. The first constraint makes the representation align together with the corresponding concept dimension, and the second one maintains the vertical parent-child relationship between concepts. To this end, the optimization problem can be formulated as:

$$
\begin{aligned}
\max_{\mathbf{q}_1, \cdots \mathbf{q}_{N_c}} \sum_{i=1}^{N_c} [\frac{1}{n_i} \mathbf{q}_i^T \psi(\mathbf{Z}^{C_i}) \mathbf{1}_{n_i \times 1} + \\
\sum_{c \in \{C_{i.children}\}} \frac{1}{n_i \times N_{cd}} (\mathbf{q}_c)^T \psi(\mathbf{Z}^{C_i}) \mathbf{1}_{n_i \times 1}], \\
s.t. \quad \mathbf{Q}^T \mathbf{Q} = \mathbf{I}_d,
\end{aligned}
\tag{3.2}
$$

where $N_{cd} = |\{C_{i.children}\}|$ is the number of child concepts of C_i. To solve this optimization problem with the orthogonality constraint, a gradient descent method with the curvilinear search algorithm (Wen & Yin, 2013) is adopted. With the whitening matrix $\mathbf{W}$ and rotation orthogonality matrix $\mathbf{Q}$, HaST-CW can replace any batch normalization layer in deep neural networks. The details of representation whitening for HaST-CW is summarized in Algorithm 1.

The overall training pipeline of our HaST-CW model is shown in Algorithm 2. We adopt an alternative training scheme. In the first stage, the deep neural network is trained with the traditional classification loss. In the second stage, we solve for $\mathbf{Q}$ to align representation dimension with semantic concepts. The two stages work alternatively during the training process. The classification loss of the first stage is defined as:

$$
\min_{\theta, \omega, \mathbf{W}, \mu,} \frac{1}{m} \sum_{i=1}^{m} \ell(g(\mathbf{Q}^T \psi(\Phi(\mathbf{x}_i; \theta); \mathbf{W}, \mu); \omega); y_i), \tag{3.3}
$$

where $\Phi(\cdot)$ and $g(\cdot)$ are layers before and after the HaST-CW module parameterized by θ and ω, respectively. $\psi(\cdot)$ is the whitening transformation parameterized by the sample mean μ and whitening matrix $\mathbf{W}$. The rotation orthogonal matrix $\mathbf{Q}$ will be updated according to Eq. (3.2) in the second stage. The operation of $\mathbf{Q}^T\psi(\cdot)$ forms the HCW module. During the first training stage, $\mathbf{Q}$ will be fixed and other parameters $(\theta, \omega, \mathbf{W}, \mu)$ will be optimized according to

Algorithm 1 Forward Pass of HCW Module

```
Input: mini-batch input $\mathbf{Z} \in \mathbb{R}^{d \times n}$
Optimization Variables: orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$
Output: whitened representations $\hat{\mathbf{Z}} \in \mathbb{R}^{d \times n}$
The batch mean: $\mu = \frac{1}{n}\mathbf{Z} \cdot \mathbf{1}$
The centered representations: $\mathbf{Z_C} = \mathbf{Z} - \mu \cdot \mathbf{1}^T$
Calculate ZCA-whitening matrix $\mathbf{W}$
Calculate the whitened representation: $\hat{\mathbf{Z}} = \mathbf{Q}^T \mathbf{W} \mathbf{Z_C}$
```

Eq. (3.3) to minimize the classification error. The first stage will take T_{thre} mini batches (we set $T_{thre} = 30$ in experiments). After that, $\mathbf{Q}$ will be updated by the Cayley transform (Wen & Yin, 2013):

$$\mathbf{Q}' = (\mathbf{I} + \frac{\eta}{2}\mathbf{A})^{-1}(\mathbf{I} - \frac{\eta}{2}\mathbf{A})\mathbf{Q}, \tag{3.4}$$

$$\mathbf{A} = \mathbf{G}\mathbf{Q}^T - \mathbf{Q}\mathbf{G}^T, \tag{3.5}$$

where $\mathbf{A}$ is a skew-symmetric matrix. $\mathbf{G}$ is the gradient of the concept alignment loss, which is defined in Algorithm 2. η is the learning rate. At the end of the second stage, an updated $\mathbf{Q}'$ will participate in the first training stage of the next iteration.

3.4.4 Semantic Constraint Loss

Besides preserving the vertical parent-child relationship of concepts, we further model the horizontal relation between concepts that are at the same hierarchy level (i.e., brothers or cousins). Different from the HCW in Eq. (3.2) that focuses on concept alignment, here we directly control the distance between representations of different concepts with the horizontal relation (Chopra et al., 2005; Schroff et al., 2015). To this end, we propose a Semantic Constraint (SC) loss

Algorithm 2 The Overall Framework of HaST-CW

```
Input: Training dataset $\mathcal{D}_T = \{\mathbf{x}_i, y_i\}_{i=1}^n$, Concept datasets $\mathcal{D}_C = \{\mathbf{X}^{C_1}, \mathbf{X}^{C_2}, \ldots, \mathbf{X}^{C_{N_c}}\}$
Optimization Variables: $\mathbf{W}, \mathbf{Q}, \theta, \mu, \omega, \mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_{N_c}]$
Hyperparameters: $\beta, \eta$
for $t = 1$ $to$ $T$ do
  Randomly sample a mini-batch$\{\mathbf{x}_i, y_i\}_{i=1}^n$from $\mathcal{D}_T$
  Do one step of SGD w.r.t $\theta$ and $\omega$ on the loss $\frac{1}{n}\sum_{i=1}^n \ell(g(\mathbf{Q}^T\psi(\Phi(\mathbf{x}_i;\theta);\mathbf{W},\mu);\omega);y_i)$
  Update $\mathbf{W}$ and $\mu$ by exponential moving average
  if $t$ mod $T_{thre}$ = 0 then
    for $i \in \{1, 2, \ldots, N_c\}$ do
      Sample mini-batches $\{\mathbf{x}_j^{C_i}, y_j\}_{j=1}^{n_i}$ from $\mathcal{D}_C$
      $\mathbf{g}_i = -\frac{1}{n_i}\sum_{j=1}^{n_i}\psi(\Phi(\mathbf{x}_j^{C_i};\theta);\mathbf{W},\mu)$
      $C_{child} \in \{C_{i.children}\}$,and $child \in \{1, 2, \ldots, N_c\}$
      $N_{child} = |\{C_{i.children}\}|$
      for $child$ do
        Sample mini-batches $\{\mathbf{x}_j^{C_{child}}, y_j\}_{j=1}^{n_{child}}$
        $\mathbf{g}_{child} = -\frac{1}{n_{child} \times N_{child}}\sum_{j=1}^{n_{child}}\psi(\Phi(\mathbf{x}_j^{C_{child}};\theta);W,\mu)$
      end for
    end for
  end if
end for
```

to model the horizontal brother-cousin relationship as below:

$$\mathcal{L}_{SC} = \alpha\mathcal{L}_{\mathcal{B}} + \beta\mathcal{L}_{\mathcal{C}}, \tag{3.6}$$

$$\mathcal{L}_{\mathcal{B}} = \sum_j \sum_{\mathcal{B}_i \in \{C_{i.\mathcal{B}}\}} \sum_k max\{0, m_{\mathcal{B}} - d(\mathbf{z}_j^{C_i}, \mathbf{z}_k^{\mathcal{B}_i})\},$$

$$\mathcal{L}_{\mathcal{C}} = \sum_j \sum_{\mathcal{B}_i \in \{C_{i.\mathcal{B}}\}} \sum_{\mathcal{C}_i \in \{C_{i.\mathcal{C}}\}} \sum_k \sum_l max\{0, d(\mathbf{z}_j^{C_i}, \mathbf{z}_k^{\mathcal{B}_i}) - d(\mathbf{z}_j^{C_i}, \mathbf{z}_l^{\mathcal{C}_i}) + m_{\mathcal{C}}\}.$$

There are two components in the SC loss and their contributions are controlled by two hyperparameters – α and β. The first term $\mathcal{L}_{\mathcal{B}}$ is a contrastive loss, which takes a pair of image representations labeled by two brother concepts as input and enlarges the distance between them. It uses a hyperparameter $m_{\mathcal{B}}$ to control the distance. The distance between two concepts increases when $m_{\mathcal{B}}$ is set larger. $\mathcal{B}_i \in \{C_{i.\mathcal{B}}\}$ denotes one of the brothers of concept C_i. The second term $\mathcal{L}_{\mathcal{C}}$ is a triplet loss. It takes three inputs: the anchor image representation $\mathbf{z}_j^{C_i}$, the image representation $\mathbf{z}_k^{\mathcal{B}_i}$ labeled by brother concept of the anchor, and the image representation $\mathbf{z}_l^{\mathcal{C}_i}$ labeled by cousin concept of the anchor. $\mathcal{C}_i \in \{C_{i.\mathcal{C}}\}$ denotes the cousins of concept C_i. The triplet loss encourages the anchor-brother distance to be smaller compared with the anchor-cousin distance in representation space. In this way, the distance of image representations from brother classes tends to be smaller than the distance of image representations from cousin classes. The gap between the two types of distance is controlled by the margin value $m_{\mathcal{C}}$. Consequently, the hierarchical concept whitening module, together with the SC loss, enables the latent representations of concepts with similar semantics to be close with each other in the latent space.

3.4.5 Latent Feature Maps Activation

The proposed HaST-CW model can generate latent representations ($\hat{\mathbf{z}}_i$) for input images ($\mathbf{x}_i$) at each neural network layer by $\hat{\mathbf{z}}_i = \mathbf{Q}^T \psi(\Phi(\mathbf{x}_i;\theta);\mathbf{W},\mu)$. The latent representation can be used to assess the interpretability of the learning process by measuring the degree of activation of $\hat{\mathbf{z}}_i$ at different concept dimensions (i.e. $\{\mathbf{q}_i\}$). In the implementation, $\Phi(\cdot)$ is a CNN based deep network, whose convolution output $\mathbf{z}_i = \Phi(\mathbf{x}_i;\theta)$ is a tensor with the dimension $\mathbf{z}_i \in R^{d\times h\times w}$. Since $\hat{\mathbf{z}}_i$ is calculated by $\hat{\mathbf{z}}_i = \mathbf{Q}^T\psi(\mathbf{z}_i)$ where $\mathbf{Q}^T \in R^{d\times d}$, we obtain $\hat{\mathbf{z}}_i \in R^{d\times h\times w}$, where d is the channel dimension and $h \times w$ is the feature map dimension. The hierarchical concept whitening operation $\mathbf{Q}^T\psi(\cdot)$ is conducted upon the d feature maps. Therefore, different feature maps contain the information of whether and where the concept patterns exist in the image. However, as a tensor the feature map cannot directly measure the degree of *concept activation*. To solve this problem and at the same time to reserve both of the high-level and low-level information, we first apply the max pooling on the feature map and then use the mean value of the downsteam feature map to represent the original one. By this way, we reshape the original feature map $\mathbf{z}_i \in R^{d\times h\times w}$ to $\mathbf{z}'_i \in R^{d\times 1}$. Finally, $\mathbf{z}'_i$ is used to measure the activation of image $\mathbf{x}_i$ at each concept dimension.

3.5 Experiments

In the experiments section, we first visually demonstrate how our method can effectively learn and hierarchically organize concepts in the latent space (Sec. 3.5.2).

We also show that (Sec. 3.5.3), compared to existing concept whitening methods, HaST-CW not only separates concepts, but also can separate groups of semantically related concepts in the latent space. After that, we discuss the advantages offered by our method with quantitative results and intuitive examples (Sec. 3.5.4) compared with baselines, including the CW module and ablated versions of our method.

3.5.1 Experimental Setting

Data Preparation

In this work, we use a newly collected and curated Agri-ImageNet dataset to evaluate the proposed HaST-CW model. In total, 9173 images from 21 classes are used in our experiments. Each image is labeled with the class at the highest possible hierarchy level. For example, an image of Melrose apple will be labeled as "Melrose" rather than the superclass "Apple". Then we divide images per class into three parts by 60%/20%/20% for a standardized training/validation/test splitting. Because the resolution of the original images can range from 300 to 5000, we adopt the following steps to normalize the image data: 1) we first lock aspect ratio and resize the images to make the short edge to be 256; 2) During each training epoch, the images in the training and validation datasets are randomly cropped into 224×224; 3) During testing process, images in the test dataset are center cropped to be of size 224×224; 4) After cropping, the pixel values of images are normalized to [0,1]. Then, the whole training dataset is divided into two parts ($\mathcal{D}_T$ and $\mathcal{D}_C$ in Algorithm 2). $\mathcal{D}_C$ is the concept dataset used to update the matrix $\mathbf{Q}$ in the second stage (Eq. (3.4)). It is created by randomly selecting 64 images from each class in the training dataset. The remaining images in the training dataset $\mathcal{D}_T$ are used in the first stage to train the model parameters (Eq. (3.3)).

Model Setting

In this work, we use several ResNet structures (K. He et al., 2016a) to extract features from images, including ResNet18 and ResNet50. During the training process, the two-stage training scheme adopts a 30-to-1 ratio to alternatively train the whole framework. In this case, after 30 mini batches of continuous training, the model will pause and the rotation orthogonal matrix $\mathbf{Q}$ will be optimized at the next mini batch. Two hyper-parameters α and β in the SC loss are set to be 1.0. Adam optimizer is used to train the whole model with a learning rate of 0.1, a batch size of 64, a weight decay of 0.01, and a momentum rate of 0.9.

3.5.2 Visualization of Semantic Map

To illustrate the learned semantic hierarchical structure, we show the representations extracted from the latent hidden layer of all the samples in Figure 3.4. For better visualization, we use Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) to project the representations to a two-dimensional space. All the images are color coded using the 17 sub-concepts which are defined on the left of Figure 3.4. The top panel shows the result using CW method. In general, all the concepts are assembled as small groups, but neither semantic relations nor hierarchical structures have been learned. We highlight the super-concept of "Weed" (black) and three sub-concepts ("Apple Golden" - green, "Apple Fuji" - red and "Apple Melrose" - blue) in the right column. We can see that the three types of apple (sub-concepts) are evenly distributed along with other fruits samples. The bottom panel shows our HaST-CW results. All the different concepts successfully keep their distinct cluster patterns as CW result. After our two-stage training process to instill the semantic and hierarchical knowledge, the three types of apple images have been pulled together and form a new concept ("Apple" with orange circle) at a higher level. Moreover, the newly learned concept of "Apple" simultaneously possesses sufficient distance to "Weed" (different super-concept) and maintains relatively close relations to "Strawberry", "Orange", "Mango" as well as other types of "Fruit". This result demonstrates the effectiveness of our hierarchical semantic concept learning framework, without negatively affecting the overall classification performance.

3.5.3 Efficiency and Accuracy of Concept Alignment

In this section, we compare the learning efficiency and accuracy of the proposed HaST-CW with that of the conventional CW method. We track the alignment between image representations and their corresponding concepts at each layer. Specifically, we randomly select six concepts, and for each concept we sort and select the top five images whose representations show the strongest activation at the corresponding concept axis. We show the results at both shallow and deep layers (layer 4 vs. layer 8) in Figure 3.5. From the results of layer 4 (the left column) we can see that most of the top five images obtained by conventional CW (the rows marked by green box) are mismatched with the corresponding concepts. For example, the five images under the concept of "Apple-Melrose" obtained by CW are from the "Weed" class. The five images under the concept of "Snake Weed" are actually from other subclass of "Weed". Moreover, this situation continues in the following layers and has not been changed until layer

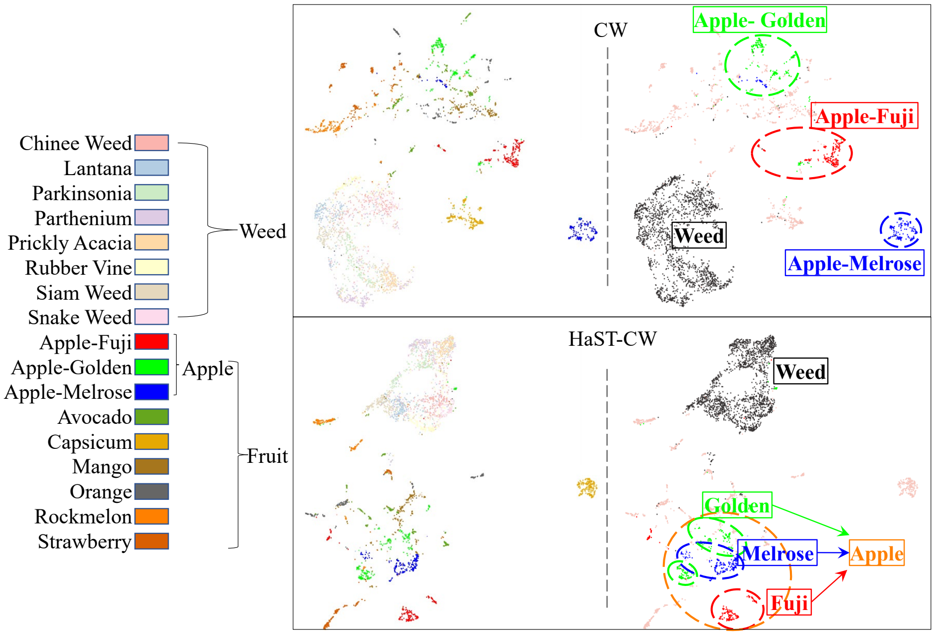


Figure 3.4: UMAP visualization of the latent space embedding with Agri-ImageNet images, colored according to the legend of image labels on the left. The top panel shows the results of the CW method and we highlight the super-concept "Weed" (black) and three sub-concepts. As shown in the bottom panel, we apply the same rules to the output of HaST-CW and visualize the results. In addition, we draw an orange circle that encapsulates three types of apples to represent the super-concept "Apple".

8. On the contrary, with the help of our designed semantic constraint loss, our HaST-CW (the rows marked by orange boxes) can learn the intrinsic concept faster and achieves the best performance at an earlier training stage (e.g., at a shallow layer). This result demonstrates that by paralleling multiple HCW layers the proposed HaST-CW model can capture the high-level features more efficiently.

To further demonstrate the alignment between images and the corresponding concepts, we project each image in the test dataset into a latent space where each concept can be represented by an axis. To visualize the alignments at different concept hierarchies (Figure 3.2), we show three pairs of concepts which belong to different hierarchical levels as examples: "Apple-Melrose"-"Apple-Fuji" is from hierarchy 3 (H-3), "Snake Weed"-"Parkinsonia" is from hierarchy 2 (H-2), and "Weed"-"Apple" crosses hierarchies 1 and 2 ("Weed": H-1, "Apple": H-2).

Within each concept pair, a two-dimensional space has been built by taking the two concepts as axes. Thus, each image can be mapped into the space by calculating the similarity between image representation and the two concept representations. The results are shown in Figure 3.6. Different rows correspond to different methods and the concept axes (space) are defined at the bottom.

The first column of Figure 3.6 shows the data distribution in the two-dimensional space of "Apple-Melrose"-"Apple-Fuji" concept pair. The images belonging to Apple-Melrose class should have the highest similarity with the concept of "Apple-Melrose", and thereby they should be located at the right-bottom corner. Similarly, the images of Apple-Fuji class should be located at the left-top corner. The other images should distribute in the space according to the similarity with the two concepts. For example, compared to images of fruit-related classes, images of weed-related classes will have lower semantic similarity with the two concepts, so they should locate near the origin point (left-bottom corner). As shown in the first column, the two models which adopt the HaST-CW method (the second and third rows) can better follow the above-mentioned patterns. While in the CW model (the first row), nearly all the images are gathered at the right-bottom corner. This may be due to the high similarity between the two concepts considered, since they share the same super-class of "Apple". As a result, CW model may be limited in distinguishing different classes with high semantic similarity. A similar situation happens in the second column with the concept pair of "Snake Weed"-"Parkinsonia". These results suggest that compared to CW method, HaST-CW can better capture the subtle differences of semantic-related classes.

The third column shows the results of the concept pair of two super-classes: "Weed" and "Apple". As each of the super-class concept contains multiple subclasses, the intra-class variability is greater. Our proposed HaST-CW, together with the SC loss (the third row), can effectively capture the common visual features and project the "Weed" and "Apple" images to the left-top and right-bottom, respectively. At the same time, the images belonging to different subclasses under "Weed" and "Apple" are assembled as blocks instead of scattered along the diagonal line. In the other two methods, especially in the CW method (the first row), the images of "Weed" class spread out over a wide range along the vertical axis. This result suggests that the proposed HaST-CW with SC loss can effectively model both the inter- and intra- class similarity.

3.5.4 Interpretable Image Classification

In this section, we compare the classification performance of the proposed HaST-CW method and the SC loss function with the conventional CW method using different backbones: ResNet18 and ResNet50. The results are summarized in Table 3.1. Different rows correspond to different model settings. Within each model setting, we repeat the experiments for five times to reduce the effect of random noise. The mean and variance of accuracy (ACC.) are reported in the fourth column. From the results, we can see that the classification performance is slightly better than the other three model settings. This result indicates that the proposed HaST-CW model can improve the interpretability without hurting predictive performance.

Table 3.1: Comparison of Classification Performance.

Module	Backbone	Loss	Acc.
CW	ResNet18	$\mathcal{L}_{CE}$	63.48 ± 0.68
CW	ResNet50	$\mathcal{L}_{CE}$	69.25 ± 3.93
HaST-CW	ResNet50	$\mathcal{L}_{CE}$	69.30 ± 3.75
HaST-CW	ResNet50	$\mathcal{L}_{CE} + \mathcal{L}_{SC}$	$\mathbf{69.49} \pm \mathbf{3.20}$

To track and visualize the classification process, we randomly select two images from Apple-Melrose class and Snake Weed class. The activation values between each image with the six relevant concepts are calculated and normalized to [0, 1]. The images, concepts and activation values are organized into a hierarchical activation tree. The results are shown in Figure 3.7. We could observe that the activation values of each image correctly represent the semantic relationship between the images and the concepts. For example, in Figure 3.7 (a), the image located at the root is from Snake Weed class which is a subclass of Weed. The activation values of the image are consistent with this relationship and possess the highest activation values on the two concepts – “Weed" and “Snake Weed".

3.6 Conclusion and Future Work

In this study, we propose a new HaST-CW and demonstrate its superiority over Concept Whitening (Z. Chen et al., 2020). HaST-CW decorrelates representations in the latent space and aligns concepts with corresponding dimensions. In addition, it correctly groups concepts at different granularity levels in the latent space and preserves hierarchical structures of concepts of interest. By doing so, we can interpret concepts better and observe the semantic relationships among concepts. We believe there are many possibilities for future work. One promising direction is automatically learning concepts from data. In this scenario, we can jointly learn possible concepts from common abstract features among images and how to represent these learned concepts in the latent space. In addition, HaST-CW can be extended with post-hoc interpretability strategies (such as saliency-based methods that highlight focused areas used for classification). In general, given the increasing demand of interpretability in deep learning, our work complements previous work and lays a solid foundation for further exploration.

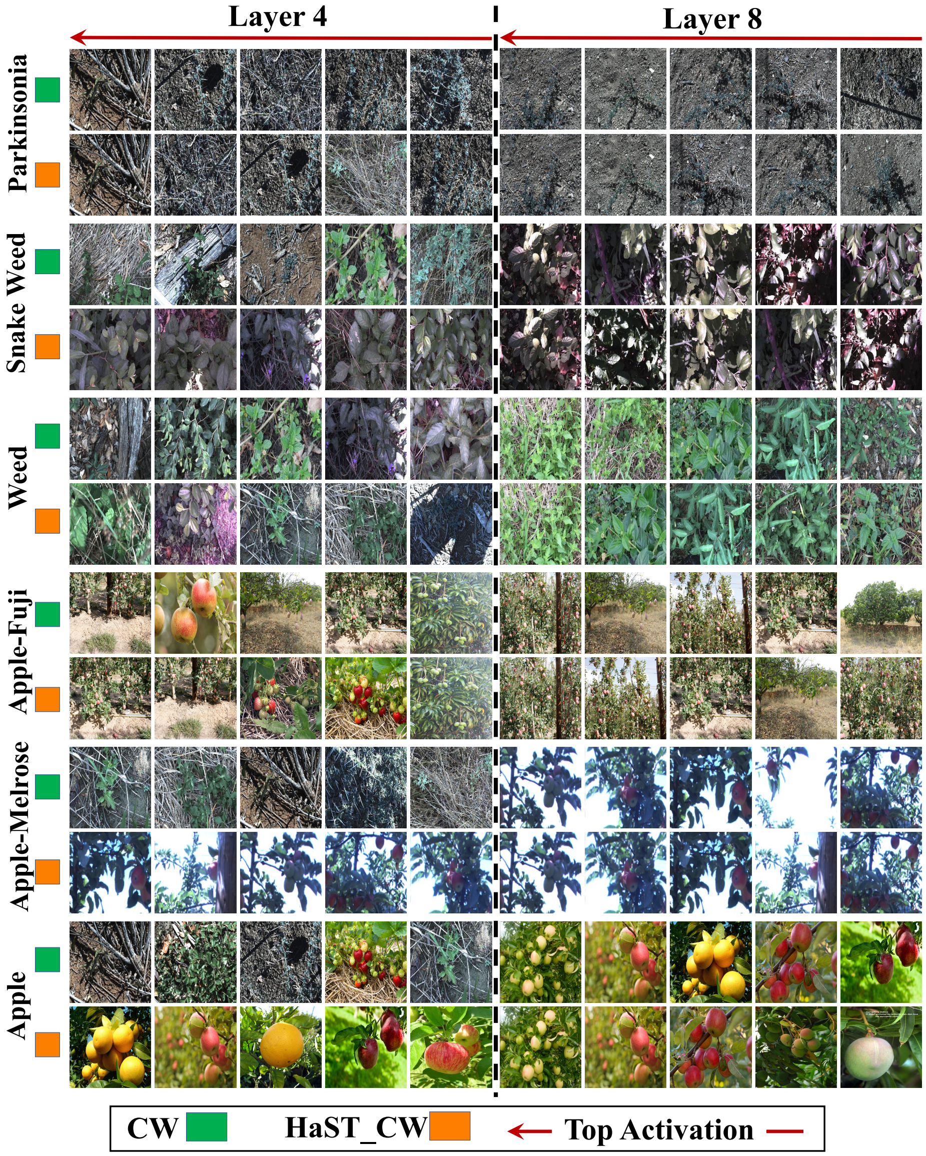


Figure 3.5: Top 5 activation images of each concept. The image panel is divided into two sets of columns: the left set of columns contains the results of layer 4 (a shallow layer), whereas the right set of columns holds the results of layer 8 (a deeper layer). Each concept covers two rows that correspond to the results of the conventional CW (marked by green boxes) and the proposed HaST-CW (marked by orange boxes), respectively.

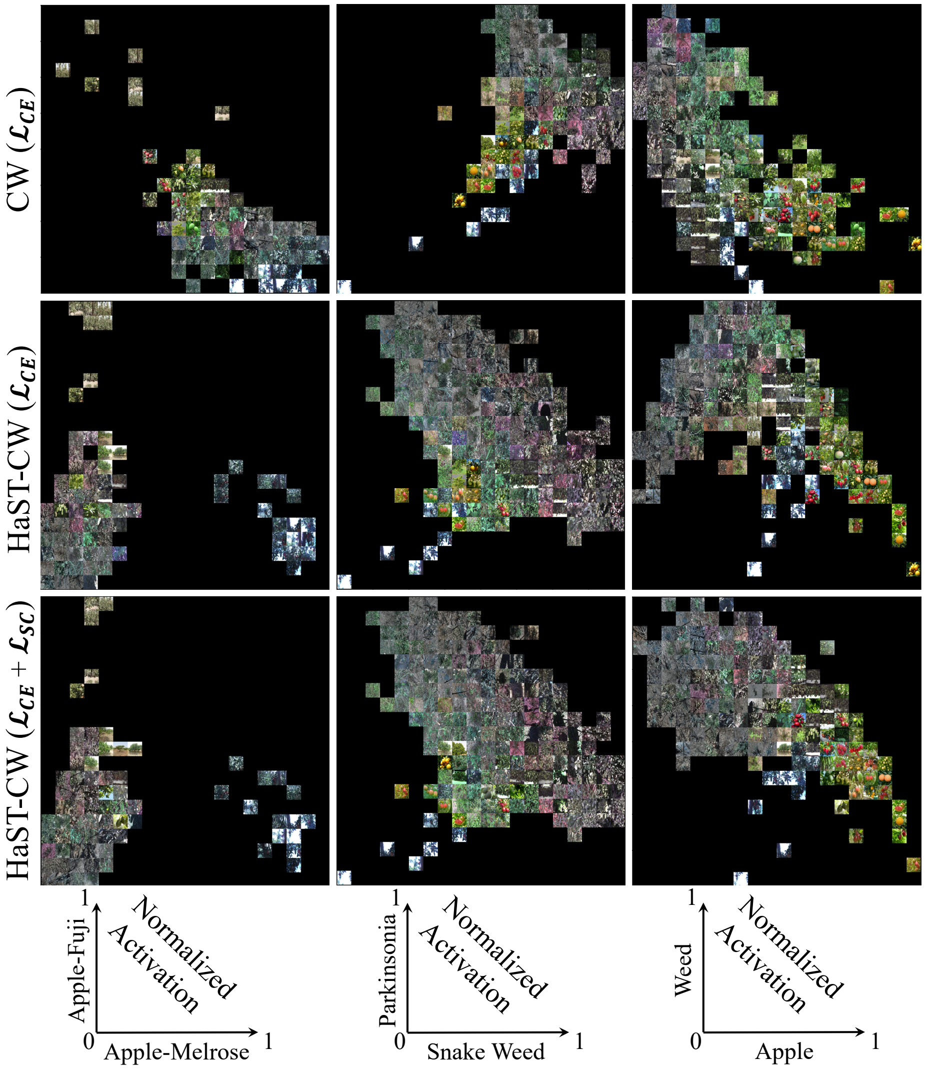


Figure 3.6: Data distribution in the concept latent space. Three pairs of concepts corresponding to different semantic hierarchy levels are selected. For each concept pair, a two-dimensional space is built by taking the concepts as axes. To visualize the alignments between images and the concepts, the images are projected into the two-dimensional space by similarity values between image representations and the two concept representations. Different rows in the figure panel correspond to different methods and the concept axes (space) are defined at the bottom.

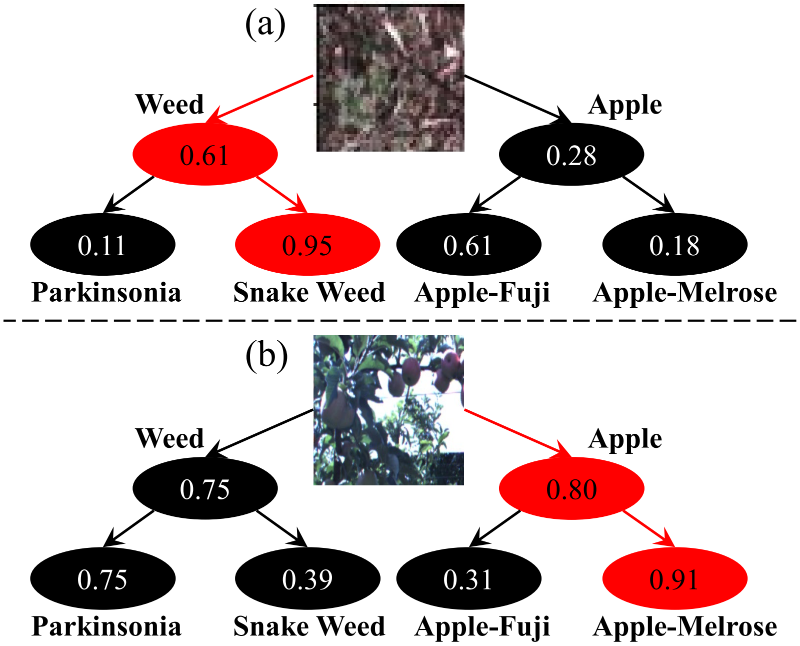


Figure 3.7: Hierarchical activation tree. We randomly select two images from the Apple-Melrose class and the Snake Weed class. For each image, activation values corresponding to the 6 concepts are calculated and normalized to [0, 1]. The highest activation values are highlighted with red along the hierarchical path.

Chapter 4

Core Pereiphery Structure

4.1 Gyri vs. Sulci: Disentangling Brain Core-Periphery Functional Networks via Twin-Transformer

4.1.1 Overview

The human cerebral cortex is highly convoluted into convex gyri and concave sulci. It has been demonstrated that gyri and sulci are significantly different in their anatomy, connectivity and function: besides exhibiting opposite shape patterns, long-distance axonal fibers connected to gyri are significantly denser than those connected to sulci, and neural signals on gyri are more complex in the low-frequency band while sulci have more complex patterns in the high-frequency band. Although accumulating evidence shows significant differences between gyri and sulci, their primary roles in brain function have not been elucidated yet. To solve this fundamental problem, we design a novel Twin-Transformer framework to explore and unveil the unique functional roles of gyri and sulci as well as their relationship and interaction in the whole brain function. Our Twin-Transformer framework adopts two identical and connected (twin) Transformers to model and disentangle spatial-temporal patterns of gyri and sulci: one focuses on the information of gyri and the other is on sulci. The Gyro-Sulcal interactions, along with the tremendous but widely existing variability across individuals, are characterized and represented via a novel Gyro-Sulcal Commonality-Variability Disentangled Loss (GS-CV Loss). We validated our Twin-Transformer on one of the largest brain imaging datasets (HCP task-fMRI gray-ordinate dataset), for the first time, to elucidate the different roles of gyri and sulci in brain function. Our results suggest that gyri and sulci could work together in a core-periphery network manner, that is, gyri could serve as core networks for information gathering and distributing in a global manner, while sulci could serve as periphery networks for specific local information processing. These findings have shed new light on our fundamental understanding of the brain's basic structural and functional mechanisms.

4.1.2 Background

The human cerebral cortex (top of Fig. 4.1-a) is highly convoluted into convex gyri and concave sulci (Fig. 4.1-b). Gyri and sulci serve as the basic building blocks to make up complex cortical folding patterns, and are fundamental to realize the brain's basic structural and functional mechanisms. Numerous efforts have been devoted to understanding the function-anatomy patterns of gyri and sulci from various perspectives, including genetics (Richiardi et al., 2015), cell biology (Gertz & Kriegstein, 2015), and neuroimaging (H. Liu et al., 2019a). It has been demonstrated consistently that gyri and sulci are significantly different in their anatomy, connectivity and function. Several studies (Fischl & Dale, 2000; Hilgetag & Barbas, 2005; G. Li et al., 2015; J. Nie et al., 2012) found that the formation of gyri/sulci may be closely related to the micro-structure of white matters. For example, diffusion tensor imaging (DTI) derived long-distance axonal fibers connected to gyri are significantly denser than those connected to sulci (bottom of Fig. 4.1-a). That is, the long-distance fiber terminations dominantly concentrate on gyri rather than sulci, and interestingly, this phenomenon is evolutionarily preserved across different primate species. Meanwhile, using functional magnetic resonance imaging (fMRI), a few functional measurements that can directly reflect brain functional activities on gyri and sulci have been explored, such as functional BOLD signals (H. Liu et al., 2019a), correlation-based connectivity/interaction (Deng et al., 2014), and spatial distribution of functional networks (J. Lv, Jiang, Li, Zhu, Chen, et al., 2015; J. Lv et al., 2014). Despite accumulating functional differences found between gyri and sulci, their basic roles as well as their relationship and interaction in the whole brain function have not been explored or elucidated yet.

To answer this fundamental question in brain science, we proposed a novel Twin-Transformer framework (Fig. 4.1-c) to explore and unveil the unique functional roles of gyri and sulci. Unlike traditional factorization-based approaches that assume linearity and independence, the Transformer attention mechanism is an ideal backbone to characterize, represent and reveal the complex and deeply buried patterns in the observed brain functional data. Our whole framework is illustrated in Fig. 4.2. Our Twin-Transformer framework adopts two identical and connected (twin) Transformers to model and disentangle spatial-temporal patterns of gyri and sulci: one focuses on the information of gyri and the other focuses on sulci. To model the complex 4D (spatial-temporal) fMRI data, within each transformer, we designed a spatial module and a temporal module to disentangle and extract the patterns in both spatial and temporal domains from the original fMRI signals. The two Transformers are connected and interact via a group of shared weights and constraints between the two spatial/temporal modules. In addition, to effectively capture the Gyro-Sulcal interactions, as well as the tremendous and widely existing variability across individual brains, a novel Gyro-Sulcal Commonality-Variability Disentangled Loss (GS-CV Loss) is proposed to guide the training process. After the model is well-trained, the functional brain networks (FBNs) and the corresponding temporal activations that are specific to gyri and sulci can be recovered by the corresponding transformers. We validated our Twin-Transformer on the one of the largest brain imaging datasets (HCP task-fMRI gray-ordinate dataset), for the first time, to elucidate the different roles of gyri and sulci in brain function. Our results suggest that gyri and sulci could work together in a core-periphery network manner (Fig. 4.1-d), that is, gyri could serve as core networks for information gathering and distributing in a global manner, while sulci could serve as periphery networks for specific local information processing. These findings have shed new light on our fundamental understanding of the brain's basic structural and functional mechanisms. The contributions of this paper are summarized as follows:

- We introduced a novel Twin-Transformer to represent and unveil the fundamental functional roles of the two basic cortical folding patterns: gyri and sulci.
- We discovered unique functional role patterns that are specifically located on gyri (global) and sulci (local).
- We found that gyri and sulci may work together in a Core-Periphery network manner: gyri serve as core networks for information gathering and distributing, while the sulci serve as periphery networks for specific local information processing.

4.1.3 Related Works

Gyri and Sulci

Gyri and sulci are the standard morphological and anatomical nomenclature of cerebral cortex and are usually defined in anatomical domains (Jiang et al., 2021).

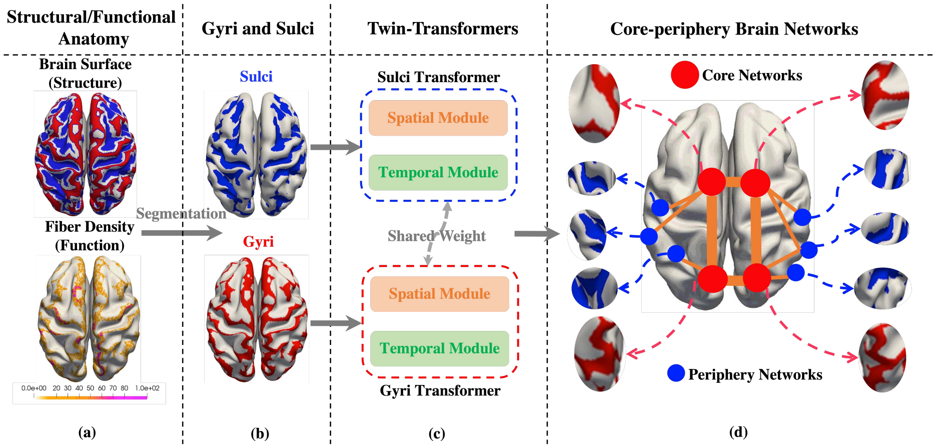


Figure 4.1: Core-periphery brain networks in gyri and sulci. (a) is the brain structural and functional anatomy of gyri and sulci. (b) is the segmentation of gyri and sulci. (c) is the proposed Twin-Transformer, where one is for gyri and the other is for sulci. (d) is the core-periphery brain networks derived from the gyri and sulci, where gyri is the core network, and sulci is the periphery network.

Neuroscientific studies have demonstrated that gyri and sulci may emerge from a complex cortical folding process, which is closely related to neurodevelopment (G. Li et al., 2015), cytoarchitecture (Fischl et al., 2008), and cognitive functioning (Honey et al., 2010). Moreover, specific gyral-sulcal patterns have been widely reported to be closely relevant to brain neuronal processes (De Juan Romero & Borrell, 2015; Johnson et al., 2015), functional activity (Troiani et al., 2020), and human behaviors (Yang et al., 2019). Therefore, gyral-sulcal patterns play important roles in brain anatomy, function, and cognition. Unveiling their fundamental roles as well as their relationship and interaction in the whole brain function is of fundamental importance to understand the underlying structural and functional brain mechanisms. In this paper, for the first time, we proposed a novel Twin-Transformer framework to elucidate the different roles of gyri and sulci in brain function.

Transformer

Since it was first proposed in 2017 (Vaswani et al., 2017), with its strong representation capacity, transformer and its variants, such as BERT (Devlin et al., 2018) and Generative Pre-trained Transformer (GPT) (Brown et al., 2020), have achieved breakthroughs in the NLP domain. Inspired by the tremendous success of transformer architectures in NLP, vision transformer (ViT) (Dosovitskiy et al., 2020) has been proposed by introducing transformer architecture into image representation learning and utilized to address a variety of vision tasks, such as image classification (M. Chen et al., 2020), object detection (Carion et al., 2020), semantic segmentation (Zheng et al., 2021), image processing (H. Chen et al., 2021), and video understanding (L. Zhou et al., 2018). Thanks to its exceptional performance, transformer-based vision models have become a potential alternative to CNN in image processing domain. To leverage the brilliant spatial and temporal representation capacity of ViT in handling image/video data, we proposed a novel Twin-Transformer framework to capture the complex gyral-sulcal spatial-temporal patterns from brain function data.

Core-periphery Network

Core-periphery (M. P. Rombach et al., 2014) structures are widely existing in transportation systems (Roth et al., 2012), social networks (Boyd et al., 2006), financial networks (**haldane2011systemic**), and brain networks (Guillon et al., 2019). The study (Guillon et al., 2019) on brain complex network reported the core/periphery networks in region of interest (ROI) with fMRI, MEG and DWI data. Another study (S. Gu et al., 2020) demonstrated the core-periphery network universally exists in human functional brain networks and unified the core-periphery with the modular organization. However, existing studies are limited in simply reporting core-periphery structure may exist in brain newtork, the factor behind this biological phenomena is unclear. In this work, using our novel Twin-Transformer model we are able to unveil that gyri and sulci, as the two basic anatomical folding patterns, serve as the core network and periphery network, respectively.

4.1.4 Methods

Gyri and Sulci Data Preparation

In our experiments, we used high-quality task-based fMRI (tfMRI) data of 540 subjects from the Human Connectome Project (HCP), that is, 3 Tesla motor and working memory (WM) task gray-ordinate dataset (H. Liu et al., 2019b) (**Barch2013**). The publicly available preprocessed tfMRI data went through the minimal preprocessing pipelines that are especially designed for high spatial and temporal resolution of HCP datasets (**Glasser2013**). The preprocessed tfMRI imaging data is a kind of 4D imaging data, which consists of a time-series of 3D images of the brain. For motor task-fMRI, each voxel contains a series of brain signals of length 284. We reorganize the signals in each voxel into a 2D matrix. In this way, a 4D tfMRI imaging can be represented by a 2D matrix, where rows represent the tfMRI time series and columns represent the brain voxels (Fig. 4.2-a). We normalized the brain signals to zero mean and unit variance. Since each subject of the preprocessed data has 59,412 voxels in standard grayordinate space, the column dimension of the 2D matrix is 59,412. To facilitate patch partition, we expanded the space dimension from 59,415 to 60,000 by adding zero vectors along the spatial dimension. Finally, a set of 2D brain signal matrices of all the subjects with dimensions of 284×60,000 are generated. Then we map the gyri and sulci masks onto the 2D brain signal matrix of each subject, and both gyri and sulci signal matrices of the 284×60,000 are generated correspondingly.

Twin-Transformers

To reveal the common and variable patterns contained in the gyri and sulci, a novel Twin-Transformer framework is proposed, including a gyri transformer and a sulci transformer. The architecture of the Twin-Transformer is illustrated in Fig. 4.2. There is a spatial and temporal self-attention module in the gyri transformer for disentangling spatial and temporal patterns of gyri as shown in Fig. 4.2-c. The structure of the sulci transformer is the same as the gyri transformer. For each input signal matrix, spatial patches are generated by shifting window along the space dimension, as illustrated by the orange arrow in Fig. 4.2-a, while temporal patches are generated by shifting window along the time dimension, as shown in the green arrow in Fig. 4.2-a. Gyri transformer generates spatial and temporal patterns of brain networks on gyri, while sulci transformer generates spatial and temporal patterns of those on sulci. By constraining the spatial and temporal patterns between gyri and sulci, commonality and variability between gyri and sulci can be discovered.

Specifically, within gyri or sulci transformer, the spatial self-attention module is designed to learn the latent representations of spatial features, and it focuses on the space dimension and takes non-overlapping spatial patches as tokens to build attention across the spatial variant patches and generate spatial patterns. It divides the input signal matrix into P non-overlapping patches by shifting the sliding window (orange dotted box following orange arrow) from left to right along the space dimension. The size of the sliding window can be adjusted according to the size of the input data. Each spatial patch contains complete temporal information of the focal brain region. The P patches correspond to P components of brain networks as predefined. Patches are used as tokens, and each token is first fed into a linear projection layer to obtain the latent representation $z_i \in \Re^{1\times D_1}$ and then the learnable spatial positional embedding, $E_i^s \in \Re^{1\times D_1}$ is added to the representations of each input token. The

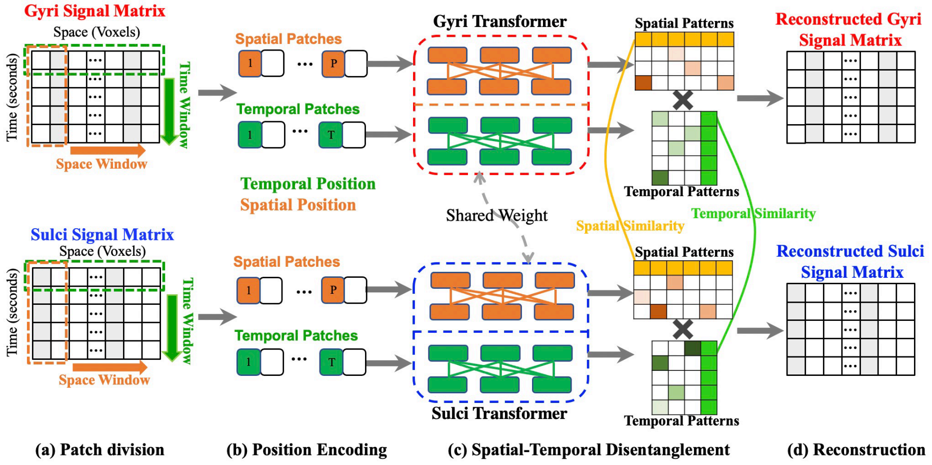


Figure 4.2: Illustration of the proposed Twin-Transformer framework. (a) shows the patch division of the gyri and sulci signal matrices. (b) is the position encoding for the spatial and temporal patches. (c) shows the details of the Twin-Transformer. The gyri transformer shares weights with sulci transformer, and each transformer includes a spatial and temporal self-attention module for processing spatial patches and temporal patches. (d) is the reconstruction of the gyri and sulci signal matrices from disentangled spatial and temporal patterns.

spatial transformer encoder can be formulated as:

$$Spa(Z) = MLP(MSA(LN(z_1^S || z_2^S || z_3^S || ... || z_P^S))) \tag{4.1}$$

where MSA() is the multi-head self-attention, MLP() represents multilayer perceptron, and LN() is layernorm. $z_i^s = (z_i + E_i^S), i = 1, 2, ..., P$ and $||$ denotes the stack operation. $Spa(Z) \in P \times N$ is the output of the spatial Transformer, where P represents the number of brain networks and N is the number of voxels in the brain. $Spa(Z)$ models the activated voxels within each brain network.

The temporal transformer is designed to learn the latent representations of temporal patterns of brain networks. The temporal self-attention module focuses on the temporal dimension and the non-overlapping temporal patches are used as tokens. Correspondingly, the temporal Transformer builds attention across the temporal variant patches and generates temporal features. Similar to the spatial transformer, by shifting the sliding window (green dotted box following green arrow) from top to bottom along the time dimension, T non-overlapping temporal patches are generated. The size of the sliding window equals 1, hence the number of patches equals the length of the brain signals.

Each temporal patch contains information of all the voxels. After input embedding and positional embedding, each patch is represented by $z_i^t = (z_i + E_i^t)$, $i = 1, 2, ..., T$. The temporal self-attention module can be formulated as:

$$Tem(Z) = MLP(MSA(LN(z_1^t||z_2^t||z_3^t||...||z_P^t))) \tag{4.2}$$

The outputs $Tem(Z)$ of the temporal self-attention module have a dimension of $Tem(Z) \in T \times P$, where T represents the time points of the fMRI signals. $Tem(Z)$ represents the signal pattern of each brain network. Taking $Spa(Z)$ and $Tem(Z)$ together, we can obtain both the spatial and temporal patterns of each pair of gyri and sulci.

Gyri and Sulci Commonality-Variability Disentangled Loss

To simultaneously capture common and variable patterns in the gyri and sulci , a new gyri-sulci commonality-variability disentangled loss (GS-CV Loss) is proposed. There are three components in GS-CV Loss. The first one is the signal matrix reconstruction loss. The whole framework is trained in a self-supervised manner to reconstruct the input signal matrix from the learned spatial and temporal patterns of gyri and sulci. This is crucial to ensure the learned spatial and temporal features can capture the complete spatial and temporal information of the input data. The reconstruction loss can be formulated as:

$$L_{reco} = \sum \|X - Spa(Z) \cdot Tem(Z)\|_{L1} \tag{4.3}$$

where X is the input signal matrix, and we use L1-norm to constrain the reconstruction of the input gyri and sulci pair. The second component is the commonality constrain loss of spatial patterns between gyri and sulci, which aims to find the common spatial patterns between gyri and sulci. For this purpose, the learned spatial feature matrix is divided into common part (the first p rows) and variable part (the remaining rows). The common and variable patterns can be learned by minimizing the difference between common parts of gyri and sulci and leaving the variable parts to learn freely. This can be formulated as:

$$L_{comm\_spa} = \sum Corr(\|Spa(Z_1)[-p :, *] - Spa(Z_2)[-p :, *]\|) \tag{4.4}$$

where $[0 : p, *]$ represents the first p rows in $Spa(Z_i)$, and $\star$ means for each row, all the elements in the columns are included, and vice versa. Since the scale of the brain signals in gyri and sulci is different, we adopt the Pearson correlation coefficient to constrain the similarity of common spatial patterns between gyri and sulci to be maximized. Similarly, the commonality constraint on temporal features, which is the third component in GS-CV Loss, is formulated as:

$$L_{comm\_tem} = \sum Corr(\|Tem(Z_1)[*, 0:p] - Spa(Z_2)[*, 0:p]\|) \quad (4.5)$$

In order to make spatial patterns distinct and limit the scale of temporal pattern from being arbitrarily large, we add a normalization on temporal features, which is formulated as:

$$L_{tem\_norm} = max(0, \frac{1}{P}(\sum_{i=1}^{P} \|Tem(Z_i[*, i])\|_2) - 1) \quad (4.6)$$

Combining the four parts, the GS-CV Loss can be formulated as:

$$GS - CV\_Loss = \alpha L_{reco} + \beta L_{comm_{spa}} + \gamma L_{comm_{t}em} + \delta L_{tem\_norm} \quad (4.7)$$

where the regularization parameters α, β, γ, and δ controls the balance of different factors on the overall loss function.

4.1.5 Results

We applied our method to one of the largest brain image dataset - HCP tfMRI data (we used both motor and working memory tasks in this work). Using the fMRI signals from gyri and sulci for each subject, as a paired input for Twin-Transformer, we generated the gyri/sulci related patterns: the output of each transformer includes 100 well-trained spatial components that can be interpreted as 100 FBNs that are specific to gyri and/or sulci. The corresponding 100 temporal components can be treated as the representative signals of each FBN in the embedding space. We first illustrate the global/local patterns of gyri/sulci using both individual and group-wise results, revealing that gyri and sulci may work together in a Core-Periphery network manner. To examine the core-periphery concept in temporal domain, we further analyze the task involved rate (TIR) of the temporal components. We found that gyri have much higher TIR than sulci, which indicates that gyri participate more in tasks than sulci do. In addition, gyri dominant FBNs show clearly global distribution patterns, while sulci dominant FBNs display an opposite local mode. All of these results taken together suggest that gyri serve as core networks for information gathering and distributing, while the sulci serve as periphery networks for specific local information processing. We also tested our proposed methods on another tfMRI dataset of working memory, and the conclusions are reproducible and consistent, and these results can be found in supplementary material.

Core-Periphery Network

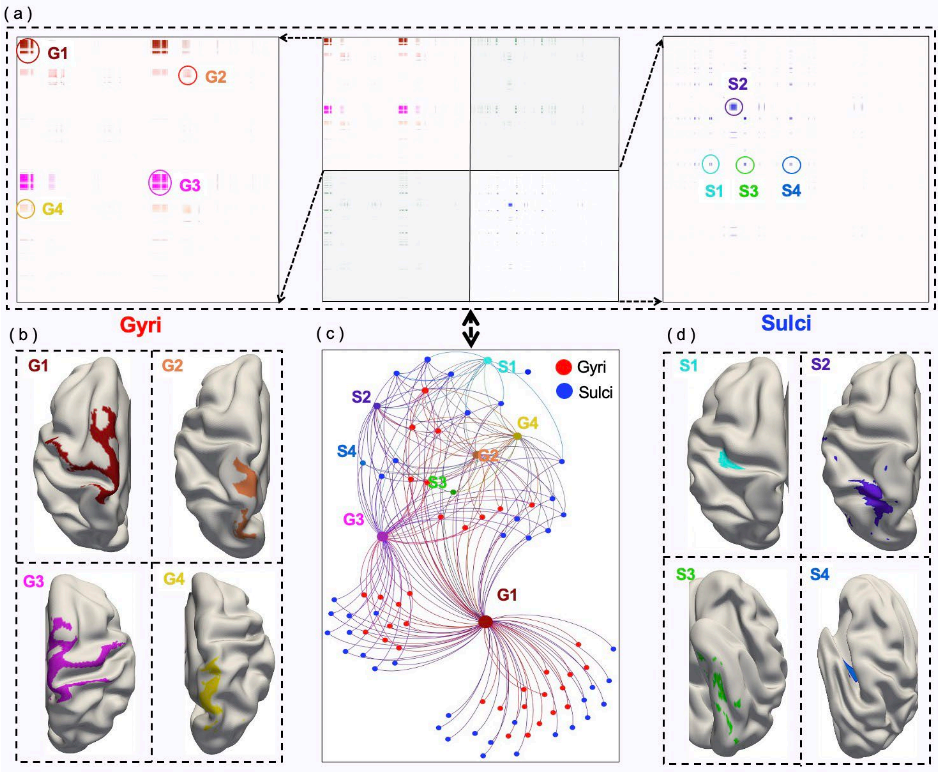


Figure 4.3: Core-Periphery Relationship Between Gyri and Sulci. (a): The activated voxels within gyri and sulci in common spatial brain networks. For better visualization, we enlarge the gyri and sulci parts into the left one and the right one. The major clusters of activated voxels in gyri are marked as G1-G4, whereas the major clusters in sulci noted as S1-S4. (b): The brain regions that correspond to the major activated brain voxels in gyri. The notations and colors are consistent with the gyri part in (a). (d): The brain regions that correspond to the major activated brain voxels in sulci. The notations and colors are consistent with the sulci part in (a). (c): Connected graph of the entire relationship matrix in (a). The red points are gyri, and the blue points are sulci.

We can identify the activated brain voxels whose weights are consistently above a pre-defined threshold across all gyri- or sulci- derived spatial components. By connecting all the activated brain voxels, we construct a relationship matrix of gyri and sulci. Fig. 4.3 shows an example of one randomly selected subject (more individual cases and group-wise results have been included in supplementary material). There are 17,232 voxels for gyri and 18,327 voxels for sulci in this subject's gray-ordinate surface, so the dimension of the obtained relationship matrix is 35559×35559, and 17232×17232 and 18327×18327 for gyri part and sulci part, respectively. The middle in Fig. 4.3-a demonstrates the entire relationship matrix, the sub-figures on left and right highlight the connections within gyri and sulci voxels, which are located in the top-left and bottom-right of the relationship matrix. In general, the relation matrix is sparse, which means only a few regions (voxels) are involved in a specific task at the same time, and this result is consistent with previous literature reports (H. Huang et al., 2017; Q. Li et al., 2021; H. Liu et al., 2019b; J. Lv, Jiang, Li, Zhu, Chen, et al., 2015). The most interesting finding using our Twin-Transformer is that the activated brain voxels in gyri-gyri section (left in Fig. 4.3-a) incline to form larger and connected blocks or clusters, as highlighted with four circles (G1-G4), while the activated brain voxels in sulci-sulci section (right in Fig. 4.3-a) tend to assemble as much smaller and scattered patterns (S1-S4). It worth noting that if the voxels are close in relationship matrix, they also tend to be neighbors on cortical surface. Therefore, after mapping the blocks of G1-G4 to cortex, we can see large continuous gyri regions on the brain surface (Fig. 4.3-b) forming gyri-based FBNs. However, the activated regions of sulci (sulci-based FBNs) are relatively small and separated (Fig. 4.3-d). To further examine the relationship between gyri-based and sulci-based FBNs, we visualize the gyri-sulci section which is located in the bottom left of the relationship matrix, as a connected graph shown in Fig. 4.3-c. We labeled the nodes in the graph with previously identified G1-G4 and S1-S4, and build their connections according to the relationship matrix. We can clearly see that all the gyri-based FBNs serve as the hub nodes, and they together compose the Core Network. Meanwhile, the sulci-based FBNs serve the supporting nodes, forming the Periphery Network. That is, the Core Networks includes gyri-based FBNs and they connect each other directly; the Periphery Networks consist of sulci-based FBNs and their communications in the entire brain network rely on the Core Network.

To further prove the concept of the Core-Periphery Network of gyri and sulci, we compute the independent probability (IP) P_{GG}, P_{SS} and P_{GS} for sub-matrices A_{GG}, A_{SS}, and A_{GS} of the entire relationship matrix, which represents the interactions within gyri vertices (Core Network), sulci vertices (Periphery Network) and between gyri, and sulci vertices (between Core and Periphery Networks). Independent probability (Cucuringu et al., 2016) is defined as the probability that there is an edge between any pairs of nodes in a given matrix, and it is an important measurement to indicate if the matrix or graph is organized as Core-Periphery pattern (Holme, 2005; M. P. Rombach et al., 2014). We set three different thresholds for edge activation to calculate the IP,

Table 4.1: The Independent Probability of Gyri Sulci Network

IP	WM			MOTOR		
	0.10	0.15	0.20	0.10	0.15	0.20
P_{GG}	0.35 ± 0.02	0.30 ± 0.02	0.07 ± 0.05	0.42 ± 0.06	0.12 ± 0.02	0.02 ± 0.01
P_{GS}	0.20 ± 0.02	0.16 ± 0.02	0.05 ± 0.05	0.37 ± 0.05	0.08 ± 0.02	0.01 ± 0.01
P_{SS}	0.12 ± 0.02	0.09 ± 0.02	0.04 ± 0.04	0.33 ± 0.05	0.06 ± 0.02	0.01 ± 0.01

Table 4.2: Gyri and Sulci Ratio Under Different Experimental Settings

Components	Comm. Spatial		Comm. Temporal		Gyri-Sulci Specific	
	Gyri Ratio	Sulci Ratio	Gyri Ratio	Sulci Ratio	Gyri Ratio	Sulci Ratio
50	52.6 ± 0.08	47.4 ± 0.08	50.1 ± 0.09	49.9 ± 0.09	50.5 ± 0.03	49.5 ± 0.03
100	53.8 ± 0.08	46.2 ± 0.08	57.5 ± 0.07	42.5 ± 0.07	51.8 ± 0.03	48.2 ± 0.03
150	54.6 ± 0.05	45.4 ± 0.05	53.2 ± 0.04	46.8 ± 0.04	51.6 ± 0.04	48.4 ± 0.04
200	54.5 ± 0.01	45.5 ± 0.01	56.7 ± 0.03	43.3 ± 0.03	55.7 ± 0.04	44.3 ± 0.04

and the average results of 500 subjects are shown in Table 1. The results show that P_{GG} > P_{GS} > P_{SS}, which confirms that our derived gyri/sulci networks have the core–periphery structure.

Task Involved Rates in Gyri and Sulci

Besides the spatial patterns of the gyri and sulci, we examine temporal patterns of gyri and sulci in this section. We calculated the Pearson correlation coefficient (PCC) between the temporal patterns and five task stimuli in motor task: left hand, right hand, left foot, right foot, and tongue. We empirically set the threshold for PCC to consider the specific temporal pattern correlated with task stimulus. We define the task-involved rates (TIR) as the number of task stimuli that the temporal patterns involved divided by the number of all stimuli. We calculated the TIR under different experimental settings, 50, 100, 150, and 200 components, and under the different thresholds for PCC. The whole TIR consists of three parts, which are common spatial TIR, common temporal TIR, and gyri-sulci specific TIR. The results are shown in Fig. 4.4. We can see that the TIR in gyri are all higher than that in sulci, except in common temporal patterns, since the common temporal patterns are trained to be similar under the temporal similarity loss. It has been widely recognized that a single brain task may need to recruit multiple brain regions or FBNs to work together. Our results show gyri have been involved more frequently, and in more tasks than sulci, which further indicates that gyri play a key role (Core Network) in brain activities, whereas sulci play a supportive role (Periphery Network).

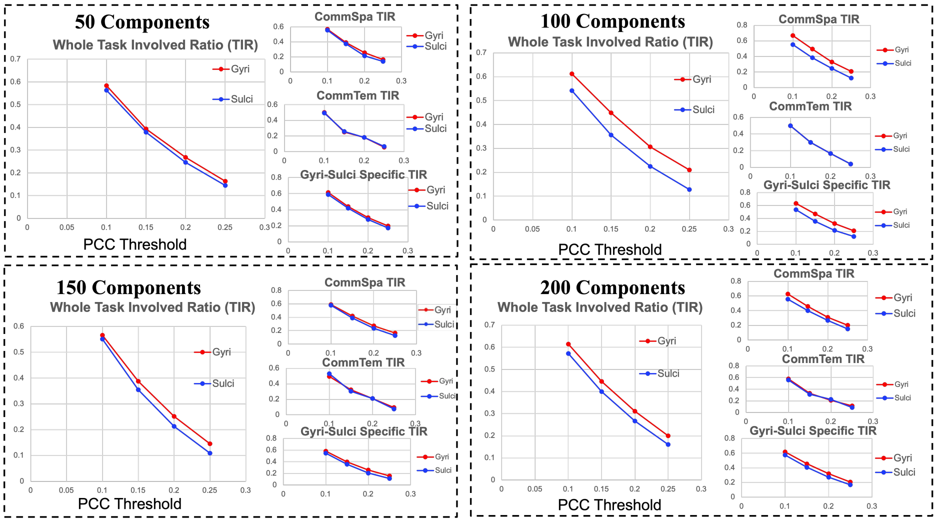


Figure 4.4: Task Involved Rates. TIR of temporal patterns of gyri and sulci. The temporal patterns are correlated with the task stimulus, and the threshold is set in the range of 0.1, 0.15, 0.2, 0.25. The four plots are the results under different experimental settings of 50, 100, 150, 200 components, where each three subplots are the detailed TIR in different parts.

Gyri/Sulci/Gyri-Sulci Dominant Network

Besides common FBNs that are derived by enforcing the external constraint, we also achieved a few FBNs that are categorized as gyri dominant (all the activated voxels belong to gyri), sulci dominant (all the activated voxels belong to sulci) and gyri-sulci collaborative brain networks (the activated voxels belong to both gyri and sulci). We display the networks of different categories from randomly selected 10 subjects in Fig. 4.5. The results are similar to the common FBNs that gyri dominant FBNs tend to have large and continuous gyri regions, while sulci dominant ones display scattered and local distributions. We also analyzed the group-wise ratio between the number of activated brain voxels in gyri and sulci using different numbers of components in our Twin-Transformer. The results are shown in Table 2. The gyri ratio is consistently higher than sulci (highlighted in bold). This result indicates that although there exist sulci dominant BFNs across subjects, the number of activated voxels in gyri is likely more than that in sulci. In summary, our proposed Twin-Transformer provides a new and powerful tool to disentangle the different functional roles of gyri and sulci with a new perspective.

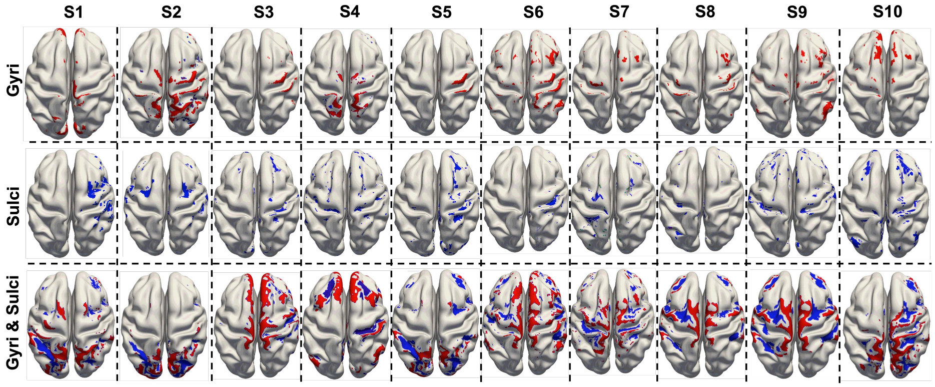


Figure 4.5: Gyri/Sulci/Gyri-Sulci Dominant Brain Networks. The three rows display of gyri dominant/sulci dominant/gyri-sulci dominant brain networks separately. They are brain functional networks from randomly selected 10 subjects.

4.1.6 Discussion

Impacts on Brain Science and Artificial Intelligence: In the brain science field, gyri and sulci are known to possess different structural, connectional and functional characteristics. However, it is the first time that our twin-transformer is powerful and accurate enough to differentiate gyri and sulci into core-periphery networks, which might suggest that the cerebral cortex is segregated into two fundamentally different functional units of gyri and sulci. This result has profound impacts on many aspects of basic, cognitive and clinical neuroscience. Core-periphery network phenomena have been reported in many real-world networked systems such as transportation, social network, financial networks, and biological neural networks, among others, and our work here revealed and characterized such core-periphery pattern in a fine-grained manner on cortical gyri and sulci. Given that the graph structures, e.g., relational graph of CNNs, of artificial neural networks in highly optimized deep learning models are more similar to those in biological neural networks, it is reasonable to postulate that the core-periphery network structure discovered in human brains in this work could be potentially infused into the design of next-generation artificial neural networks in deep learning as a prior knowledge or meaningful constraint, thus leading to brain-inspired artificial intelligence.

Limitation: Our work has several potential limitations. a) We simply add the degree from each activation graph generated by each common spatial component to build the gyri/sulci graph. There is still some room for building a better gyri/sulci graph. b) We mainly focus on discovering the relationship between gyri and sulci at this moment, and ignored the intermediate regions on the gyral wall that is between gyri and sulci. In the near future, we plan to explore the intermediate regions' roles in the core-periphery brain network system.

4.1.7 Conclusion

In this paper, we proposed a novel data-driven Twin-Transformer framework and applied it to HCP gray-ordinate tfMRI dataset to characterize the roles of cortical gyri and sulci on the brain functional networks. With this framework, we can disentangle the spatial and temporal patterns from the brain signals of gyri and sulci, providing us the possibility to quantitatively analyze the difference between gyri and sulci. The most important finding in this study is that we identified the core-periphery relationship between gyri and sulci, as well as the corresponding core-periphery brain networks. Our results show that core-periphery networks are broadly existing between gyri and sulci across all subjects. Overall, our proposed Twin-Transformer contributes to a better understanding of the roles of gyri and sulci in brain architecture, which offers new insight into the design of next-generation artificial neural networks, brain-inspired AI models, and beyond.

4.2 CORE-PERIPHERY PRINCIPLE GUIDED REDESIGN OF SELF-ATTENTION IN TRANSFORMERS

4.2.1 Overview

Designing more efficient, reliable, and explainable neural network architectures is critical to studies that are based on artificial intelligence (AI) techniques. Numerous efforts have been devoted to exploring the best structures, or structural signatures, of well-performing artificial neural networks (ANN). Previous studies, by post-hoc analysis, have found that the best-performing ANNs surprisingly resemble biological neural networks (BNN), which indicates that ANNs and BNNs may share some common principles to achieve optimal performance in either machine learning or cognitive/behavior tasks. Inspired by this phenomenon, rather than relying on post-hoc schemes, we proactively instill organizational principles of BNNs to guide the redesign of ANNs. We leverage the Core-Periphery (CP) organization, which is widely found in human brain networks, to guide the information communication mechanism in the self-attention of vision transformer (ViT) and name this novel framework as CP-ViT. In CP-ViT, the attention operation between nodes (image patches) is defined by a sparse graph with a Core-Periphery structure (CP graph), where the core nodes are redesigned and reorganized to play an integrative role and serve as a center for other periphery nodes to exchange information. In addition, a novel patch redistribution strategy enables the core nodes to screen out task-irrelevant patches, allowing them to focus on patches that are most relevant to the task. We evaluated the proposed CP-ViT on multiple public datasets, including medical image datasets (INbreast) and natural image datasets (CIFAR-10, CIFAR-100, and TinyImageNet). Interestingly, by incorporating the BNN-derived principle (CP structure) into the redesign of ViT, our CP-ViT outperforms other state-of-the-art ANNs. In general, our work advances the state of the art in three aspects: 1) This work provides novel insights for brain-inspired AI: we can utilize the principles found in BNNs to guide and improve our ANN architecture design; 2) We show that there exist sweet spots of CP graphs that lead to CP-ViTs with significantly improved performance; and 3) The core nodes in CP-ViT correspond to task-related meaningful and important image patches, which can significantly enhance the interpretability of the trained deep model. (Code is ready for release).

4.2.2 Background

Aided by the rapid advancement in hardware and massively available data, deep learning models have witnessed an explosion of various artificial neural networks (ANN) architectures (K. He et al., 2016b; Krizhevsky et al., 2017; Vaswani et al., 2017), and made breakthroughs in many application fields due to their powerful automatic feature extraction capabilities. It is widely expected the architectures of ANN, as the core of current AI techniques, to be more efficient, reliable, explainable, and transformable, to adapt to various and complex problems in real applications. Essentially, various ANN architectures, represented via different neuron wiring patterns, correspond to different information exchange mechanisms, and therefore, have an inevitable effect on the latent feature representation and the downstream task performance. For example, multilayer perceptron (MLP) directly stacks multiple layers of neurons with paired-wise full connections between adjacent layers, whereas convolutional neural networks (CNN) focus on learning effective convolutional kernels that indicate specific wiring patterns among the neurons within the receptive field. Similarly, recurrent neural networks (RNN) adopt cyclic connections between nodes, allowing output to affect subsequent input to the same nodes (Sherstinsky, 2020). This special neuron wiring pattern of building cycles between nodes also enables RNNs to model and infer temporal dynamic relationships (Tealab, 2018) contained in sequential data. More recently, transformer has become another mainstream ANN architecture due to its outstanding self-attention mechanism that allows effective and efficient message exchanges among neurons, and produced promising results in the natural language processing (Devlin et al., 2018; Vaswani et al., 2017) and computer vision domains (Dosovitskiy et al., 2020; Z. Liu et al., 2021). In particular, many advancements in transformer architecture design, e.g., vision transformer (ViT) (Dosovitskiy et al., 2020), have centered around more effective message exchange mechanisms among spatial tokens by designing different Token Mixers. For instance, the shifted window attention in Swin (Z. Liu et al., 2021), the token-mixing MLP in Mixer (Tolstikhin et al., 2021), and the pooling in MetaFormer (W. Yu et al., 2022), among others, were all designed to improve the self-attention upon the original vanilla ViT (Dosovitskiy et al., 2020), and thus enable more effective and efficient message exchanges among spatial patches/tokens. However, despite tremendous advancements in ANN architecture design in MLPs, CNNs, RNNs, and transformers, particularly for better message exchange mechanisms, there has been a fundamental lack of general principles that can inform and guide such ANN architecture design and redesign.

To seek such guiding principles for ANN architecture design, more and more research studies started exploring the "structural signatures" of well-performing ANNs. Hence, the deep learning community has witnessed a paradigm shift from optimal feature design to optimal ANN architecture design. In general, the major strategies for optimal ANN architecture design can be categorized into two basic streams based on how to search in the neural architecture space. The first strategy is to design neural architectures that achieve the best possible performance using given computing resources in an automated way with minimal human intervention. Neural architecture search (NAS) (Elsken et al., 2019; Ren et al., 2021; Zoph & Le, 2016) is a major methodology in this category. NAS has a relatively low demand for the researchers' prior knowledge and experience, making it easier to perform modifications to the neural architecture though it usually comes with a high computational cost. The second category of the strategy is to take the advantage of prior knowledge from specific domains, such as brain science, to guide ANN architecture design. For example, the authors in (Y. Zhang, Choi, et al., 2021) designed a two-stream model for grounding language learning in vision based on the brain science principle that humans learn language by grounding concepts in perception and action, and encoding "grounded semantics" for cognition. It is worth noting that the above-mentioned two strategies should be viewed as complementary to each other rather than being in conflict, and their combination provides the researchers with an opportunity to explore and design well-performing neural architectures under different principles. For instance, recent studies, via qualitatively post-hoc analysis, have found that the best-performing ANNs surprisingly resemble biological neural networks (BNN) (You et al., 2020), which indicates that ANNs and BNNs may share some common principles to achieve optimal performance in either machine learning or cognition/behavior tasks.

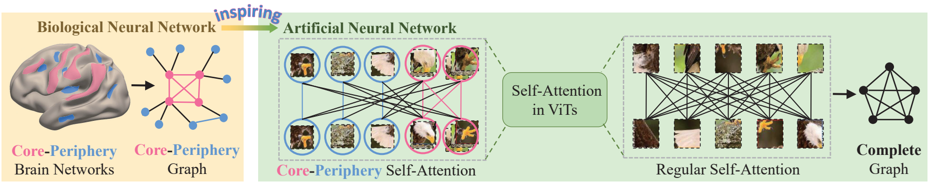

Biological Neural Network
inspiring
Artificial Neural Network
Core-Periphery Brain Networks
Core-Periphery Graph
Self-Attention in ViTs
Core-Periphery Self-Attention
Regular Self-Attention
Complete Graph

Figure 4.6: The Core-Periphery principle in brain networks inspires the design of ANNs. The Core-Periphery structure broadly exists in brain networks, with a dense "core" of nodes (pink) densely interconnected with each other and a sparse "periphery" of nodes (blue) sparsely connected to the core and among each other. Inspired by this principle of BNN, we aim to instill the Core-Periphery structure into the self-attention mechanism and propose a new CP-ViT model.

Inspired by the above-mentioned prior outstanding studies, in this work, we aim to proactively instill the Core-Periphery (CP) organization to guide the redesign of ANNs by using ViT as a working example. It has been widely confirmed that the Core-Periphery organization universally exists in the functional networks of human brains and other mammals, effectively promoting the efficiency of information transmission and communication for integrative processing (Bassett et al., 2013; S. Gu et al., 2020). The concept of the Core-Periphery brain network is illustrated in Fig. 4.6. By using the Core-Periphery property as a guiding principle, we infused its effective and efficient information communication mechanism into the redesign of ViT. To this end, we quantified the Core-Periphery property of the human brain network, infused the Core-Periphery property into ViT, and proposed a novel CP-ViT architecture. Specifically, we update the complete graph of dense connections in the original vanilla ViT (Dosovitskiy et al., 2020) with a sparse graph with Core-Periphery property (CP graph), where the core nodes are redesigned and reorganized to play an integrative role and serve as a center for other periphery nodes to exchange information. Moreover, in our design, a novel learning mechanism is used to endow the core nodes with the power to capture the task-related meaningful and important image patches. We evaluated the proposed CP-ViT on multiple public datasets, including a medical image dataset (INbreast) and natural image datasets (CIFAR-10, CIFAR-100, TinyImageNet). The results indicate that the optimized CP-ViT in sweet spots (You et al., 2020) outperforms other ViTs. We summarize our contributions in three aspects: 1) This work provides novel insights for brain-inspired AI: we can utilize the principles found in BNNs to guide and improve our ANN architecture design; 2) We show that there exist sweet spots of CP graphs that lead to CP-ViTs with significantly improved performance and 3) The core nodes in CP-ViT correspond to task-related meaningful and important image patches, which can significantly enhance the interpretability of the trained deep model.

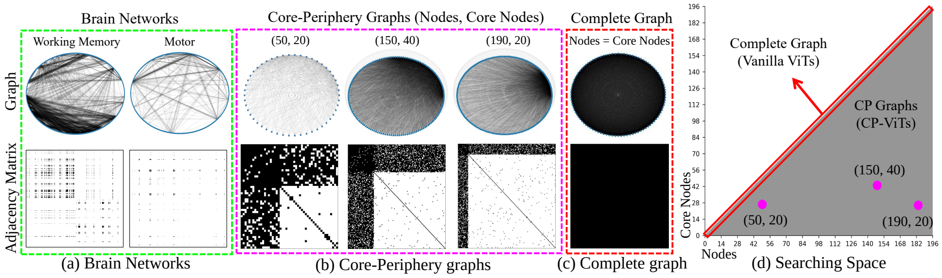


Figure 4.7: (a) Two types of representative brain networks in motor and working memory tasks. (b) Three examples of CP graphs. (c) Complete graph. The first row in (a), (b), and (c) shows their wiring patterns, while the second row shows their corresponding adjacency matrices. Black color in adjacency matrices means connections between nodes, while white represents no edge. (d) Graph search space defined by the total nodes number and the core nodes number. The complete graphs are located at the diagonal highlighted by a red box and the CP graphs are located at the remaining parts.

4.2.3 Results

Exploring Core-Periphery Graphs

Core-Periphery property in brain networks. We quantitatively measured the Core-Periphery property of brain networks. Working memory network (BN-WM) and motor network (BN-M) are two typical functional networks that are widely existed in the human brain. In this work, we used task fMRI data of these two tasks in the Human Connectome Project (Van Essen et al., 2013) to generate functional brain networks. Using voxels as nodes and the correlations between fMRI signals associated with each voxel as edges, we built two population-level functional networks and showed their connection patterns as well as the adjacency matrices in Fig. 4.7(a). To measure the Core-Periphery property of the two functional brain networks, we adopted independent probability (Cucuringu et al., 2016) as the measurement. Independent probability is defined as the probability that there is an edge between any pairs of nodes in a given matrix. Thus, the independent probabilities of the core-core connections, core-periphery connections, and periphery-periphery connections can be represented as I_{cc}, I_{cp} and I_{pp}, respectively. If the given matrix or graph is organized in a Core-Periphery manner (Holme, 2005) (M. P. Rombach et al., 2014), the corresponding independent probabilities will have the following relations: $I_{cc} > I_{cp} > I_{pp}$. According to previous studies (H. Liu et al., 2019a), the convex gyri and concave sulci areas, which are two basic anatomical structures of the cerebral cortex, play different functional roles: gyri are functional hubs for global information exchange while sulci are responsible for local information processing. Therefore, we divided the nodes (voxels) into two categories, gyri-nodes (nodes in gyri regions) and sulci-nodes (nodes in sulci regions), and examined if brain networks have CP structure: gyri-nodes act as core nodes and sulci-nodes act as periphery nodes. The core-periphery measures of brain networks are shown in the last two columns in Table 4.3. R_{cc}, R_{pp} and R_{cp} represent the normalized independent probabilities of core-core, core-periphery, and periphery-periphery connections. The independent probabilities and normalized independent probabilities are formulated as:

$$\begin{aligned} I_{cc} = \frac{1_{A_{cc}}}{\|A_{cc}\|_1}, I_{cp} = \frac{1_{A_{cp}}}{\|A_{cp}\|_1}, I_{pp} = \frac{1_{A_{pp}}}{\|A_{pp}\|_1}, \\ R_{cc} = I_{cc}/(I_{cc} + I_{cp} + I_{pp}), \\ R_{cp} = I_{cp}/(I_{cc} + I_{cp} + I_{pp}), \\ R_{pp} = I_{pp}/(I_{cc} + I_{cp} + I_{pp}). \end{aligned} \tag{4.8}$$

Core-Periphery structure in artificial neural networks. We introduced the Core-Periphery organization into ANNs by CP graphs. There are two key factors that can affect the CP graph generation process. The first is the number of nodes, including the number of total nodes and the core nodes, which defines the search space. In this work, we set the maximum number of total nodes as 196, i.e., the number of patches for the vision transformer, then the number of core nodes can be any number between 0 and 196. Thus, the search space will include $\sum_{i=1}^{196} \sum_{j}^{0<j<=i} (i + j) = 19208$ types of CP graphs, where i and j represent the number of total nodes and the core nodes. The second is the wiring patterns of CP graphs: in this work, we used p_{cc}, p_{cp}, and p_{pp}

Table 4.3: Evaluation of the Core-Periphery property in CP graphs, graphs generated by other graph generators, and brain networks

IP	CP Graphs	CE. Graphs	WS Graphs	ER Graphs	BN-M	BN-WM
R_{cc}	$.59 \pm .06$	$.33 \pm .00$	$.40 \pm .27$	$.36 \pm .23$	$.55 \pm .11$	$.61 \pm .09$
R_{cp}	$.35 \pm .13$	$.33 \pm .00$	$.40 \pm .28$	$.36 \pm .24$	$.34 \pm .07$	$.26 \pm .10$
R_{pp}	$.07 \pm .06$	$.33 \pm .00$	$.20 \pm .28$	$.28 \pm .22$	$.15 \pm .05$	$.14 \pm .06$

to represent the wiring probabilities between core-core nodes, core-periphery nodes, and periphery-periphery nodes, respectively. Fig.4.7 (b) and (c) present the wiring patterns and adjacency matrices of three examples of CP graphs and the complete graph. As shown in Fig. 4.7(b) and (c), CP graphs are densely connected for core nodes and sparsely connected for periphery nodes. The overall connection patterns of CP graphs are more sparse than the complete graph. The search space of CP graphs was shown in Fig. 4.7(d) where the complete graphs located at the diagonal were highlighted by a red box and three types of CP graphs corresponding to Fig. 4.7(b) were highlighted by pink circles. For each type of CP graph, we generated 5 samples with different wiring patterns and obtained 19208 * 5 CP graphs in total. Since the number of the generated CP graphs is huge (19208 * 5 in total), we sampled 190 types of CP graphs out of the total 19208 and finally obtained 190*5 candidates. For example, for a CP graph with 50 nodes, the number of core nodes is set to be [10, 20, 30, 40]. As a result, four different CP graphs, including [50, 10], [50, 20], [50, 30], and [50, 40], are obtained. For each of these four types of CP graphs, we generate 5 samples for further experiments.

Similar to brain networks, we also used the normalized independent probability to measure the Core-Periphery property for the generated CP graphs. We calculated the normalized averaged independent probability over 190*5 CP graphs and showed the results in the first column of Table 4.3. From the table we can see that $R_{cc} > R_{cp} > R_{pp}$, which suggests that our generated CP graphs, as expected, display prominent Core-Periphery properties, while the graphs generated by the classic graph generators, such as (1) Complete graph (CE.) generator; (2) Watts-Strogatz (WS) generator; and (3) Erdos-Renyi (ER) generator don't have the Core-Periphery property.

Table 4.4: Summary of datasets

Dataset	Training	Validation	Class	Original Res.	Resized Res.
INbreast	6000	100	3	1024 * 1024 * 3	224 * 224 * 3
CIFAR-10	50000	10000	10	32 * 32 * 3	224 * 224 * 3
CIFAR-100	50000	10000	100	32 * 32 * 3	224 * 224 * 3
TinyImageNet	100K	10000	200	64 * 64 * 3	224 * 224 * 3

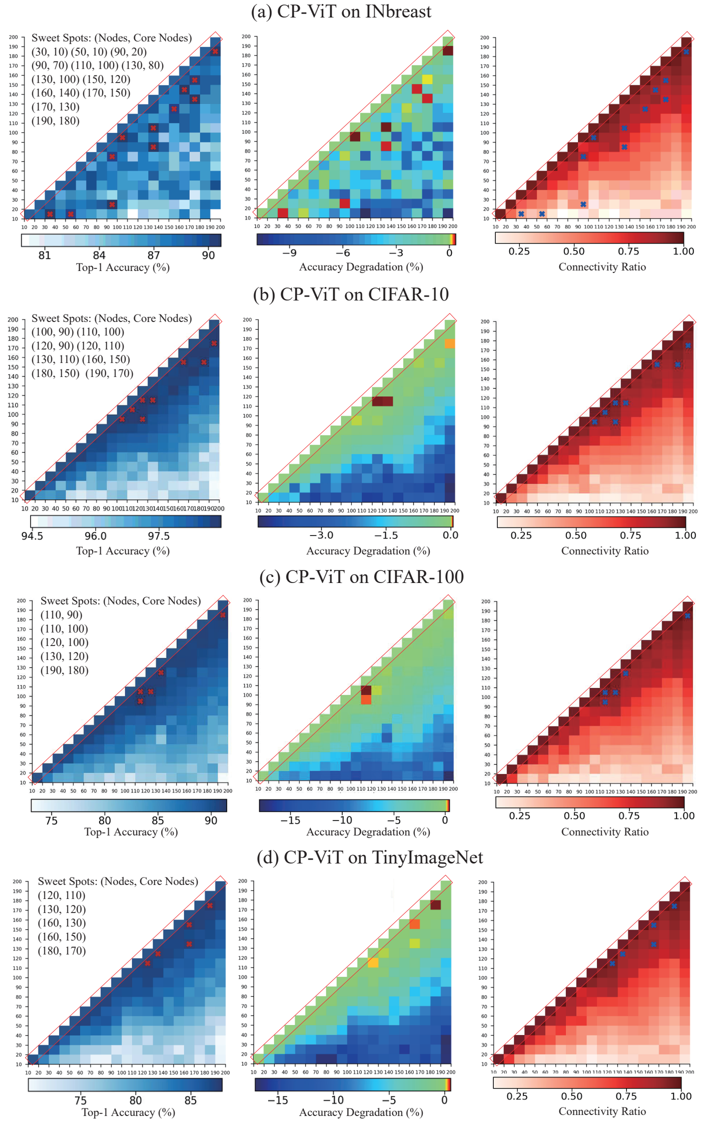
Sweet Spots for CP-ViTs

In this section, we evaluated the performance of the proposed CP-ViT. The CP-ViT was implemented based on the ViT-S/16 architecture (X. Chen et al., 2021) and evaluated on 4 different types of public datasets, the medical image dataset INbreast (Moreira et al., 2012), the natural image dataset CIFAR-10 (Krizhevsky, Hinton, et al., 2009), CIFAR-100 (Krizhevsky, Hinton, et al., 2009) and Tiny-ImageNet (Griffin et al., 2007). The summary of the datasets we used in this work is presented in Table 4.4. The parameters of CP-ViT were initialized and fine-tuned from ViT-S/16 trained on ImageNet (Krizhevsky et al., 2017). We trained the CP-ViT for 100 epochs with batch size 64 for INBreast and 256 for CIFAR-10, CIFAR-100 and TinyImageNet, and used AdamW optimizer and cosine learning rate schedule (Loshchilov & Hutter, 2016) with an initial learning rate of 0.0001 and minimum of $1e{-}6$. All the experiments were conducted using NVIDIA Tesla V100 GPU.

We explored the performance of different types of CP graphs in the search space (Fig. 4.7(a)) in terms of top 1 accuracy and connection ratio. The connection ratio (CR) quantitatively measures the computational costs of different self-attention operations, which is defined by (4.9):

$$CR = \frac{1_{M_{cp}}}{||M_{cp}||_1} \tag{4.9}$$

where $1_{M_{cp}}$ represents the number of 1s in the mask matrix of cp graphs - M_{cp} which is derived from the adjacency matrix of the CP graph, and $||\bullet||_1$ is the number of elements in the mask matrix. In general, CR represents the ratio of actual self-attention operations to the potential maximum self-attention operations. Given a graph, the potential maximum self-attention operation is fixed. Less actual self-attention operation means less computational cost and hence it has a smaller CR value.


(a) CP-ViT on INbreast

(b) CP-ViT on CIFAR-10

(c) CP-ViT on CIFAR-100

(d) CP-ViT on TinyImageNet

Figure 4.8: Performance of CP-ViT measured using INbreast, CIFAR-10, CIFAR-100 and TinyImageNet datasets. Sub-figures on the left column under each datasets show the top 1 classification accuracy of the CP-ViTs and vanilla ViTs in the search space. A deeper color means higher top 1 accuracy. Sweet spots are marked by red crosses, in which CP-ViTs achieve better performance than vanilla ViT. Sub-figures on the middle column are the accuracy degradation of the CP-ViTs compared to vanilla ViTs. Sub-figures on the right column are the self-attention connection ratio of the CP-ViTs and vanilla ViT. Lighter color means a lower connection ratio. Sweet spots are marked by the blue crosses.

Table 4.5: Comparison between the proposed CP-ViT in sweet spots with finetuned vanilla ViT-S (Dosovitskiy et al., 2020). * means vanilla ViT-S finetuned by ourselves.

Dataset	Model	CP Graph	CR (%)	R_{cc},R_{cp},R_{pp}	Top1 Acc.(%)
INbreast	ViT-S(*)	(N, N)	100.00	0.33, 0.33, 0.33	89.91
	CP-ViT	(30, 10)	32.36	0.58, 0.33, 0.09	90.58
	CP-ViT	(50, 10)	**29.20**	0.53, 0.34, 0.12	90.01
	CP-ViT	(90, 20)	43.82	0.52, 0.36, 0.12	90.58
	CP-ViT	(90, 70)	84.50	0.54, 0.40, 0.06	90.01
	CP-ViT	(100, 90)	92.80	0.49, 0.39, 0.11	**90.69**
	CP-ViT	(130, 80)	31.34	0.58, 0.34, 0.07	90.58
	CP-ViT	(130, 100)	82.94	0.57, 0.36, 0.07	**90.69**
	CP-ViT	(150, 120)	84.18	0.57, 0.41, 0.02	90.01
	CP-ViT	(160, 140)	87.77	0.55, 0.41, 0.03	90.58
	CP-ViT	(170, 130)	80.79	0.57, 0.41, 0.02	90.58
	CP-ViT	(170, 150)	87.65	0.56, 0.41, 0.03	90.12
	CP-ViT	(190, 180)	84.89	0.52, 0.42, 0.05	90.69
CIFAR-10	ViT-S(*)	(N, N)	100.00	0.33, 0.33, 0.33	98.50
	CP-ViT	(100, 90)	92.80	0.49, 0.39, 0.11	98.91
	CP-ViT	(110, 100)	94.49	0.53, 0.42, 0.05	98.91
	CP-ViT	(120, 90)	89.73	0.51, 0.41, 0.08	98.91
	CP-ViT	(120, 110)	94.70	0.49, 0.38, 0.12	98.97
	CP-ViT	(130, 110)	**87.32**	0.56, 0.40, 0.03	**98.97**
	CP-ViT	(160, 150)	90.47	0.54, 0.39, 0.06	98.91
	CP-ViT	(180, 150)	91.79	0.50, 0.42, 0.07	98.91
	CP-ViT	(190, 170)	92.59	0.53, 0.43, 0.03	98.94
CIFAR-100	ViT-S(*)	(N, N)	100.00	0.33, 0.33, 0.33	91.10
	CP-ViT	(110, 90)	88.96	0.59, 0.37, 0.04	91.32
	CP-ViT	(110, 100)	94.49	0.53, 0.42, 0.05	**91.45**
	CP-ViT	(120, 100)	92.40	0.50, 0.41, 0.09	91.15
	CP-ViT	(130, 120)	**87.50**	0.58, 0.32, 0.09	91.11
	CP-ViT	(190, 180)	94.89	0.52, 0.42, 0.05	91.12
TinyImageNet	ViT-S(*)	(N, N)	100.00	0.33, 0.33, 0.33	87.36
	CP-ViT	(120, 110)	94.71	0.49, 0.39, 0.12	87.51
	CP-ViT	(130, 120)	**87.50**	0.58, 0.33, 0.09	87.37
	CP-ViT	(160, 130)	90.02	0.54, 0.44, 0.02	87.40
	CP-ViT	(160, 150)	90.47	0.54, 0.40, 0.06	87.63
	CP-ViT	(180, 170)	95.84	0.50, 0.43, 0.07	**87.84**

For each specific combination of different numbers of nodes/core nodes in the search space, we trained the CP-ViT with 5 different CP graph samples and reported the average result in Fig. 4.8. The four results in Fig. 4.8(a-d) correspond to four different datasets. For the results on each dataset, we display three subfigures: the top 1 accuracy (left), the accuracy degradation (middle), and the connection ratio (right). We highlighted the sweet spots, which are corresponding to the CP graphs that lead to improved performance (You et al., 2020), with red crosses in Fig. 4.8. In the top-1 accuracy of Fig. 4.8, deeper color means better performance. The accuracy degradation subfigures show the accuracy variation compared to fully connected self-attention ViTs. Our CP-ViTs gain a positive boost in sweep spots as it has higher accuracy than vanilla ViTs. At the same time, our CP-ViTs maintain competitive top-1 accuracy in most search space areas, as shown in the middle subfigures. The performance of CP-ViTs varies in the search space. This result indicates that different self-attention (wiring) patterns may have great influences on the performances of ViTs. Compared to vanilla ViTs with a fully-connected self-attention pattern, the proposed CP-ViT provides the potential for the model to only search for optimal self-attention patterns. The CRs of all the ViTs including vanilla ViTs and CP-ViTs were shown on the right. The CRs of the sweet spots were marked with a blue cross. Besides the improvement in classification accuracy (0.78% for INbreast, 0.47% for CIFAR-10, 0.35% for CIFAR-100, 0.48% for TinyImageNet), the proposed CP-ViT also leads to a great reduction in connection ratio due to less self-attention operations (-70.80% connections for INbreast, -12.68% connections for CIFAR-10, -12.50% connections for CIFAR-100, -12.50% connections for TinyImageNet). The model setting, top 1 accuracy, and CRs of different ViTs were reported in Table 4.5. For all the four datasets, our CP-ViT not only shows improved classification performance but also reduces connection ratio compared to vanilla ViTs. Interestingly, our results demonstrate that the "sweet spots" are corresponding to the wiring patterns (graphs) with CP structures, instead of fully connected self-attention.

We also compared the proposed CP-ViT with the state-of-the-art methods in Table 4.6, including various convolutional networks and transformer architectures. Note that we applied the core-periphery principle to guide the design on small ViT, therefore, the counterparts we compared to in this work are also small-scale transformers and their variants. "——" means there is no available reports or not applicable. As presented in the table, our method outperforms the CNNs, and a series of variants of transformers on these datasets, suggesting the superiority of the proposed CP-ViTs over the existing methods.

Visualization of Important Patches

Another advantage of CP-ViT is that it can potentially improve the interpretability of the deep-learning models via semi-intervention when linking the explainable concepts contained in the data to the instilled CP structures (section 3.2.3).

Table 4.6: Comparisons with state-of-the-art transformers and other architectures.

Model	CIFAR-10	CIFAR-100	TinyImageNet	INbreast
ResNet-18 (K. He et al., 2016b)	95.55	76.64	67.33	84.34
ResNet-18+Gaze (S. Wang, Ouyang, et al., 2022)	−−	−−	−−	86.74
ViT-S-SAM (X. Chen et al., 2021)	98.20	87.60	87.50	90.20
ViT-S (X. Chen et al., 2021)	97.60	85.70	87.40	89.91
DeiT-S (Touvron et al., 2021)	97.50	90.30	86.90	89.90
Mixer-S-SAM (X. Chen et al., 2021)	96.10	82.40	85.60	87.60
T2T-ViT-12 (Y. Wang et al., 2021)	98.53	89.63	86.20	88.40
AutoFormer-S (M. Chen et al., 2021)	98.50	90.60	87.60	90.10
CP-ViT-S(ours)	98.97	91.45	87.84	90.69

In our CP-ViT the core nodes are expected to be associated with the important image patches relating to the classification tasks. To evaluate this, we show the patches that were redistributed to the core nodes when the model was well-trained in Fig. 4.9. For INBreast, we randomly selected the images of three subjects in each class and displayed the original images, the images overlaid with important patches, and the images overlaid with the expert's eye gazes in three columns. As shown in the Fig. 4.9, the patches of the core nodes are well co-localized with the locations that were identified as diagnostic biomarkers of the disease in literature publications (Ibrokhimov & Kang, 2022). We also show the medical physicians' eye gaze maps on these images, given that the eye gaze acquired by eye-tracking equipment is considered the ground truth for identifying important areas in the image. The important patches identified by our CP-ViT highly overlap with the eye gaze maps, demonstrating the correspondence between the core nodes and the task-related concepts, i.e., the

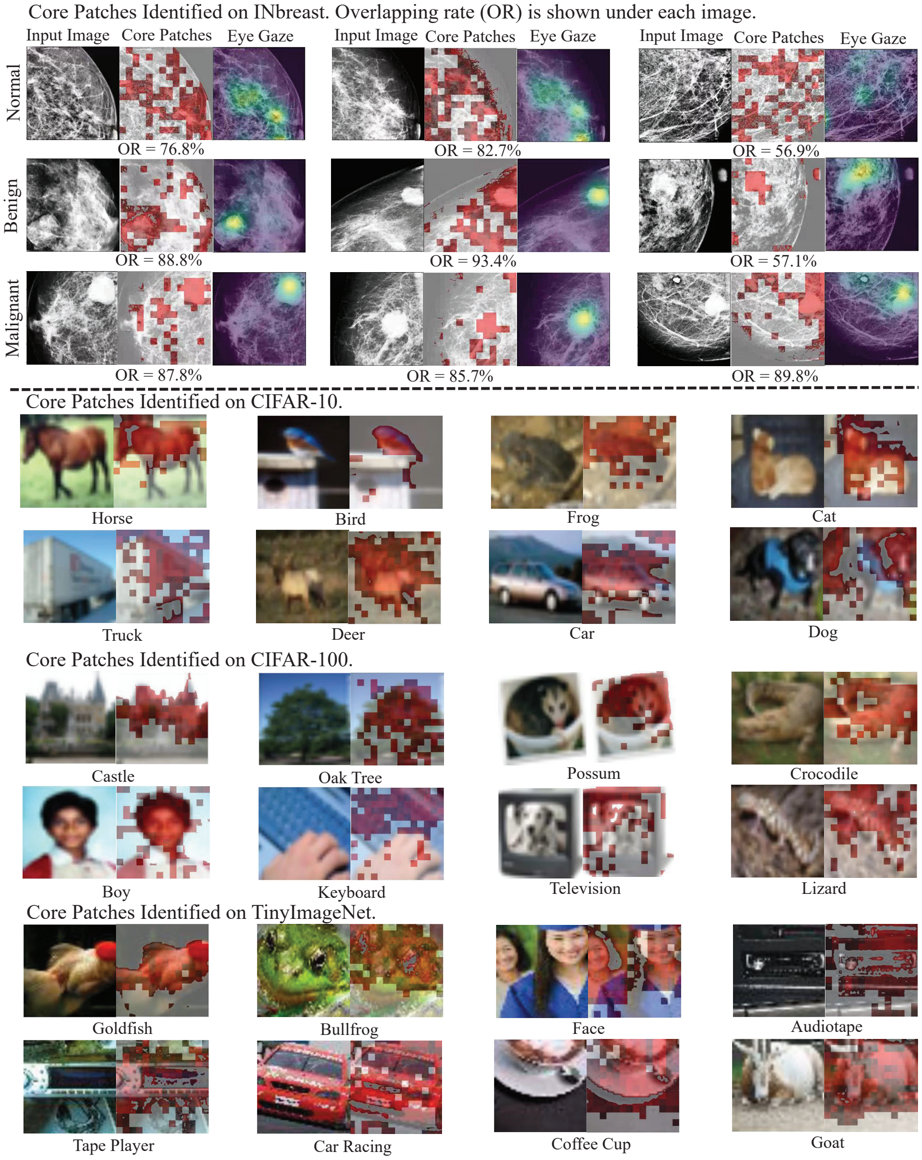
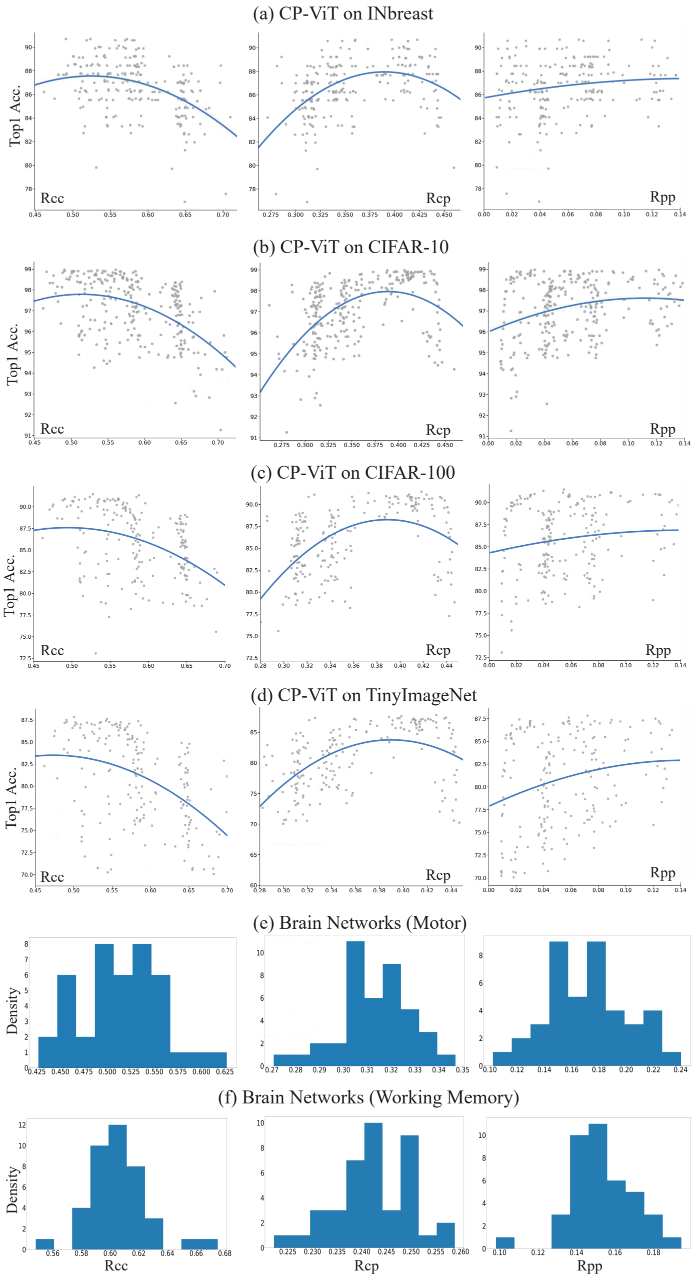


Figure 4.9: Visualization of important image patches that were distributed to the core nodes. For the INbreast dataset (the first block), images of three randomly selected subjects for each class were shown. For each subject, there are three images displayed in three columns. The left column is the original image, the middle column shows the important patches marked by red, and the right column is the eye gaze of medical physicians on the image. For the natural image datasets (the second block, CIFAR-10, CIFAR-100 and TinyImageNet), the important patches identified in eight randomly selected classes were displayed. The left column is the original image, and the right column shows the identified core patches marked in red.

important image patches. For natural image datasets, we also visualized the patches assigned to the core nodes under the black dotted line in Fig. 4.9. It is clear that the objects in the patches of core nodes are semantically related to the class labels.

Fast Search for Sweet Spots

Our proposed CP-ViT aims to achieve better performance more efficiently, by directly updating the initial dense wiring patterns with sparse CP graphs which are widely existing in BNN. Previous studies suggest that in ANN there exist sweet spots that correspond to some specific wiring patterns leading to significantly improved performance (You et al., 2020). Therefore, it is interesting to investigate the relationship between sweet spots (the ANN structures with better performance) and the introduced CP structure. We conducted intensive experiments to illustrate how the accuracy changes under the CP measurements (in terms of normalized independent probability) and the results are summarized in Fig. 4.10. We found the normalized independent probabilities between core nodes - R_{cc}, core and periphery nodes - R_{cp} and periphery nodes - R_{pp} fall in different range: $[0.45, 0.70]$ for R_{cc}, $[0.25, 0.45]$ for R_{cp}, and $[0.00, 0.15]$ for R_{cc}. Both R_{cc} and R_{cp} display obvious and consistent patterns in terms of the relationship between ANN performance (accuracy) and CP properties: there exists a certain range of CP structures with which the corresponding wiring patterns of ANN can achieve better performance. For example, when the normalized independent probabilities between core and periphery nodes (R_{cp}) fall within the range of $[0.36, 0.42]$, our CP-ViT inclines to have the best accuracy on all four datasets. On the contrary, the normalized independent probabilities between periphery nodes (R_{pp}) show relatively less influence on the overall performance. These results suggest that the wiring patterns between core nodes and periphery nodes have more influence on the overall ANN performance than the wiring patterns between periphery nodes. For comparison, we also calculated the range of group-wise normalized independent probabilities in human functional brain networks when performing two different tasks - motor and working memory tasks. The results are shown in Fig. 4.10 (e-f). Interestingly, the distribution of R_{cc}, R_{cp} and R_{cp} shows obvious overlaps among different functional brain networks though the major range of CP metrics is different from ANN (our CP-ViT). In general, our CP-ViT can leverage the CP structure to learn the optimal combinations of total nodes and core nodes, and to quickly find the sweet spots in a more efficient way.


(a) CP-ViT on INbreast

(b) CP-ViT on CIFAR-10

(c) CP-ViT on CIFAR-100

(d) CP-ViT on TinyImageNet

(e) Brain Networks (Motor)

(f) Brain Networks (Working Memory)

Figure 4.10: Visualization of Core-periphery measures versus the classification performance. The regression results of the normalized independent probability versus the classification accuracy for experiments on each dataset are presented in (a), (b), (c), and (d). The core-periphery measures for brain networks of motor and working memory are shown in (e) and (f).

4.2.4 Methods

4.2.5 Related Work

Core-periphery Structure The Core-Periphery structure is a fundamental network signature that is composed of two qualitatively distinct components: a dense "core" of nodes strongly interconnected with one another, allowing for integrative information processing to facilitate the rapid transmission of the message, and a sparse "periphery" of nodes sparsely connected to the core and among each other (Gallagher et al., 2021). The Core-Periphery pattern has helped explain a broad range of phenomena in network-related domains, including online amplification (Barberá et al., 2015), cognitive learning processes (Bassett et al., 2013), technological infrastructure organization (Alvarez-Hamelin et al., 2005; Carmi et al., 2007), and critical disease-spreading conduits (Kitsak et al., 2010). All these phenomena suggest that the Core-Periphery pattern may play a critical role to ensure the effectiveness and efficiency of information exchange within the network. In the literature, there are two widely-used approaches for generating graphs with Core-Periphery property (CP graphs): the classic two-block model of Borgatti and Everett (BE algorithm) (Borgatti & Everett, 2000) and the k-cores decomposition (Gallagher et al., 2021). The former approach partitions a network into a binary hub-and-spoke layout, while the latter one divides it into a layered hierarchy. In this work, for simplicity, we adopted a two-block model to generate a CP graph which is used to guide the self-attention operations between patches (tokens) in ViT. In this way, the Core-Periphery property is infused into the ViT model.

Methods for Designing More Efficient ViT Architecture ViT and its variants have achieved promising performances in various computer vision tasks, but their gigantic parameter counts, heavy run-time memory usage, and high computational cost become a major burden for the applications. Therefore, there is an urgent need to develop lightweight vision transformers with comparable performance and efficiency. For this purpose, several studies aimed to use network pruning, sparse training, and supernet-based NAS to slim vanilla ViT. **From token level**, Tang et al. (Tang et al., 2022) designed a patch slimming method to discard useless tokens. Evo-ViT (Y. Xu et al., 2022) updated the selected informative and uninformative tokens with different computation paths. VTP (M. Zhu et al., 2021) reduced embedding dimensionality by introducing control coefficients. **From model architecture level**, UP-ViTs (H. Yu & Wu, 2021) pruned the channels in ViTs in a unified manner, including residual connections in all the blocks, multi-head self-attention (MHSA) (Vaswani et al., 2017), feedforward neural layers (FFNs), normalization layers, and convolution layers in ViT variants. SViTE (T. Chen et al., 2021) dynamically extracted and trained sparse subnetworks instead of training the entire model. To further co-explore data and architecture sparsity, a learnable token selector was used to determine the most vital image patch embeddings in the current input sample. AutoFormer (M. Chen et al., 2021) and ViTAS (Su et al., 2021) leveraged supernet-based NAS to optimize the ViT architecture. Despite the remarkable improvements achieved by the above methods, both token-sampling and data-driven strategies may highly depend on the data and tasks performed, impeding the vision transformers' generalization capability. A more universal principle (e.g., derived from BNNs) that can guide a more efficient design of ANN's architecture is much desired. In this work, we will leverage a widely existing Core-Periphery property in BNN to develop a more efficient CP-ViT.

Core-Periphery Principle Guided Transformer

The Core-Periphery principle can be applied to ViT and its variants via a unified framework that is illustrated in Fig. 4.11. The framework includes two main parts: Core-Periphery graph generation and Core-Periphery graph guided re-design of the self-attention mechanism.

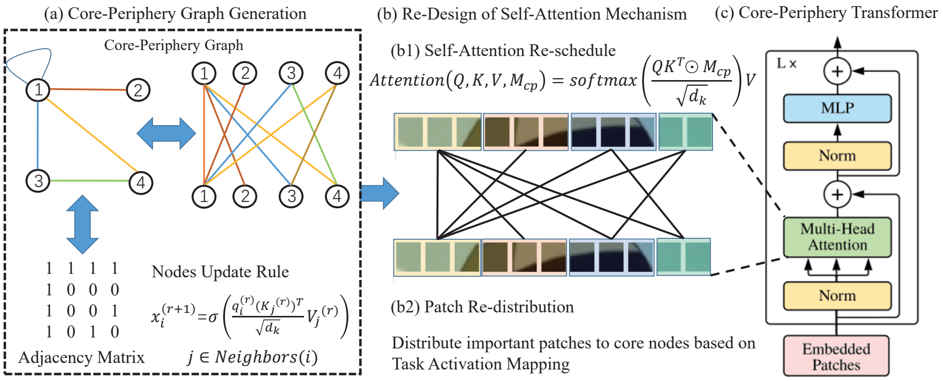

(a) Core-Periphery Graph Generation
Core-Periphery Graph
1
2
3
4
1 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
Adjacency Matrix
Nodes Update Rule
$x_i^{(r+1)} = \sigma\left(\frac{q_i^{(r)}(K_j^{(r)})^T}{\sqrt{d_k}} V_j^{(r)}\right)$
$j \in Neighbors(i)$
(b) Re-Design of Self-Attention Mechanism
(b1) Self-Attention Re-schedule
$Attention(Q, K, V, M_{cp}) = softmax\left(\frac{QK^T \odot M_{cp}}{\sqrt{d_k}}\right)V$
(b2) Patch Re-distribution
Distribute important patches to core nodes based on Task Activation Mapping
(c) Core-Periphery Transformer
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

Figure 4.11: Core-Periphery Principle Guided Re-design of Self-Attention. The proposed Core-Periphery guided re-design framework for ViTs consists of two major components: the Core-Periphery graph generator and the re-design of the self-attention mechanism. The basic idea is that we mapped the ViT structure to graphs and proposed a new graph representation paradigm to represent the self-attention mechanism. Under this paradigm, the design of the self-attention mechanism can be turned into a task of designing desirable graphs. (a) The CP graph generator was proposed to generate graphs with Core-Periphery property in a wide range of search spaces. (b) The self-attention of the nodes is controlled by the generated CP graph and the patches are re-distributed to different nodes by a novel patch distribution method. (c) The new self-attention mechanism will upgrade the regular self-attention in vanilla ViT. The new ViT architecture is thus named as CP-ViT.

Core-Periphery Graph Generation The self-attention of our proposed CP-ViT is controlled by Core-Periphery graphs (CP graphs). We proposed a CP graph generator to generate a wide spectrum of CP graphs in the graph space defined by the number of total nodes and the core nodes. Although several graph generators have been proposed in previous works, they were not designed for generating CP graphs. For example, Erdos-Renyi (ER) generator samples graphs with given node and edge numbers uniformly and randomly (Erdos, Rényi, et al., 1960); Watts-Strogatz (WS) generator generates graphs with small-world properties (Watts & Strogatz, 1998), and the complete graphs generator generates graphs where nodes are pair-wise densely connected with each other (Walker, 1992).

To generate graphs with CP property, we proposed a novel CP graph generator that is parameterized by a total node number n, a core node number m, and three wiring thresholds p_{cc}, p_{cp}, p_{pp} which are the wiring probabilities between the core-core nodes, core-periphery nodes, and periphery-periphery nodes, respectively. Based on these measures, the CP graph generation process is as follows: we first defined the core nodes number m and the periphery nodes number $n - m$; Then, for each of the core-core node pairs, we used a random seed sampled from the continuous uniform distribution in $[0, 1]$ to generate a wiring probability p_{rs}. If the wiring probability is greater than the threshold p_{cc}, the two core nodes are connected. This wiring process is formulated as:

$$A\left(i, j\right) = \begin{cases} 1 & \text{if } p_{rs} \geq p_{cc} \\ 0 & \text{if } p_{rs} < p_{cc} \end{cases} \tag{4.10}$$

where A is the adjacency matrix of the generated graph, 1 means that there exists an edge between the nodes i and j, 0 means there is no edge between the nodes. The same procedure was applied to core-periphery and periphery-periphery node pairs with the corresponding thresholds p_{cp} and p_{pp}, respectively. In this way, by using different combinations of n, m, and wiring thresholds, we can generate a large number of candidate graphs in the graph space; finally, all the generated graphs were examined by the CP detection algorithm (BE algorithm) (Borgatti & Everett, 2000) and the graphs with CP property will be used in the further steps to guide the self-attention operation.

Core-Periphery Guided Self-Attention To instill the CP principle into the self-attention mechanism in ViT, we redesigned the self-attention operations according to the generated CP graphs: the patches are replaced by the nodes, and the new self-attention relations are replaced by the edges in the CP graph. Thus, the self-attention in the vanilla ViT can be represented as a complete graph, and similarly, the CP principle can be effectively and conveniently infused into the ViT architecture by upgrading the complete graph with the generated CP graphs. CP graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with nodes set $\mathcal{V}$ and edges set $\mathcal{E}$. The redesign of self-attention is formulated as:

$$x_i^{(r+1)} = \sigma^{(r)}(\{(\frac{q_i^{(r)}(K_j^{(r)})^T}{\sqrt{d_k}})V_j^{(r)}, \forall j \in N(i)\}) \tag{4.11}$$

where $\sigma(\cdot)$ is the activation function, which is usually the softmax function in ViTs, $q_i^{(r)}$ is the query of patches in the i-th node in $\mathcal{G}$, $N(i) = \{i \| i \vee (i, j) \in \mathcal{E}\}$ are the neighborhood nodes of node i, d_k is the dimension of queries and keys, and $K_j^{(r)}$ and $V_j^{(r)}$ are the key and value of patches in node j.

In vanilla ViT, one input image is divided into 196 patches, and each patch resolution is 16 by 16. In CP-ViT, each node corresponds to a single patch or multiple patches. We proposed the following patch assignment pipeline to map the original patches to the nodes: for a CP graph with n nodes, each node will be assigned to either $\lfloor 196/n \rfloor + 1$ or $\lfloor 196/n \rfloor$ patches. For example, if we use a CP graph with 5 nodes, the 5 nodes will have 40, 39, 39, 39, and 39 patches, respectively; and if we use a CP graph with 196 nodes, each node will correspond to a single patch. Note that the patches are randomly assigned to the nodes at the beginning of the training process, and then they will be re-distributed iteratively after each training epoch based on a novel patch distribution method that will be elaborated in the next section. Based on the above discussion, the CP graph-guided self-attention conducted at the node level can be formulated as:

$$Attention(Q, K, V, M_{cp}) = softmax(\frac{QK^T \odot M_{cp}}{\sqrt{d_k}}V) \tag{4.12}$$

where the queries, keys, and values of all the patches are packed into the matrices Q, K, and V, respectively. M_{cp} is the mask matrix derived from the adjacency matrix A of the CP graph, and $\odot$ is the dot product. The size of the mask matrix M_{cp} is 197×197 (196 patches plus 1 classification token), and it is a symmetric matrix. The derivation process of M_{cp} is as follows: for a CP graph with 5 nodes, the 5 nodes have 40, 39, 39, 39, and 39 patches, respectively. If the element $(1, 2)$ in the corresponding adjacency A is 1, which means the node #1 is connecting to the node #2, and as a result, the 40 patches corresponding to the node #1 are connecting to the 39 patches associated with the node #2. Therefore, the elements at $(1:40, 40:79)$ and $(40:79, 1:40)$ in the mask matrix M_{cp} will be 1, where the $(40:79, 1:40)$ means the elements from the 40th row to 79th row, and from the 1st column to the 40th column. The elements in the last row and column of M_{cp} are 1 because the classification token is connected to all the nodes, including both core and periphery nodes. Similar to the multi-head attention in transformers (Vaswani et al., 2017), our proposed CP multi-head attention is formulated as:

$$\begin{aligned} MultiHead(Q, K, V, M_{cp}) = Concat(head_1, ..., head_h)W^o \\ where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V, M_{cp}) \end{aligned} \tag{4.13}$$

where the parameter matrices W_i^Q, W_i^K, W_i^V and W^O are the projections. Multi-head attention helps the model to jointly aggregate information from different representation subspaces at various positions. In this work, we apply the CP principle to each representation subspace.

Algorithm 3 Patch Re-Distribution

```
Input: Likelihoods of an image belonging to a particular class (before activation layer) $y$, patch embeddings $P^k, k = 1, 2, ..., 196$
Calculate the gradients of the likelihoods $y$ with respect to patch embeddings $P^k$, respectively. $\frac{\partial y}{\partial P_i^k}$.
Obtain patch important weights $\alpha_k, k = 1, 2, ..., 196$ by average-pooling of gradients over the feature dimension, $\alpha_k = \frac{1}{Z}\sum_{i=1}^{Z} \frac{\partial y}{\partial P_i^k}$, where $Z$ is the dimension of the patch embeddings.
Sort the patch important weights $\alpha_k$ in a descending manner, $Sort(\alpha_k)$.
Determine the number of patches assigned to core nodes, for simplicity, we call these patches as core patches.
Match the core patches to core nodes in a way that the patches with higher importance weights are distributed to the core nodes with a higher degree.
Re-organized the patches of the images according to the importance weights.
Output: Patch re-organized images.
```

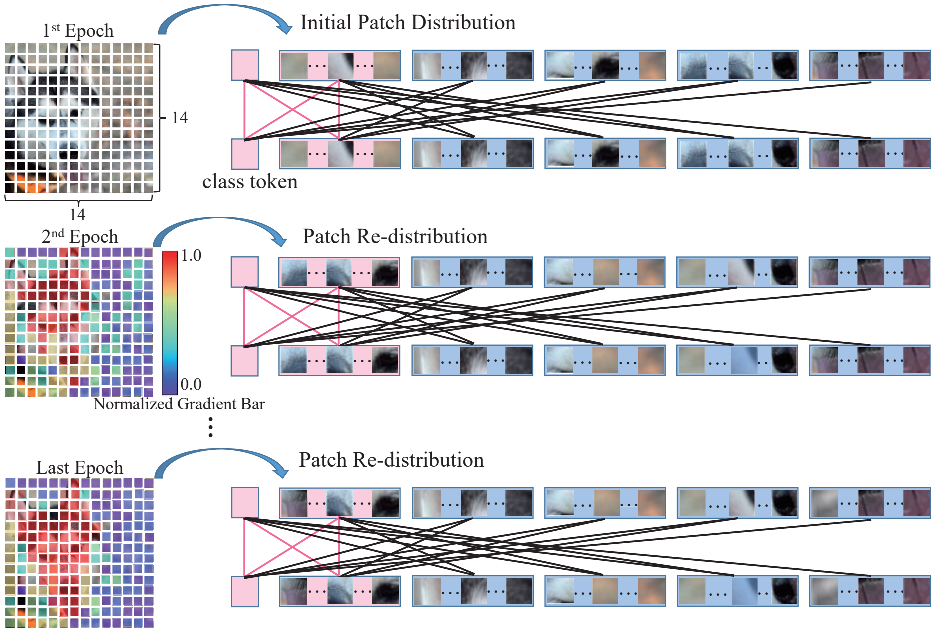


Figure 4.12: Illustration of Patch Redistribution Process. The pink nodes are the core nodes, while the blue nodes are the periphery nodes. The initial patch distribution at the first epoch is the same as the vanilla ViTs. After each iteration during the training process, the gradients of patches discriminate from each other due to different contributions to the classification. The red the image patches are, the high gradient they are. Thus, the core patches that contribute most to the classification task are re-distributed to core nodes.

Patch Redistribution The CP structure inclines to make the communication and message exchange at core nodes more intensive while less frequent among periphery nodes. This is based on the fact that the core nodes usually process the most important information in many biological networks (Bassett et al., 2013). To this end, we need to evaluate the importance of the patches and select the most important ones to assign to the core nodes, which is defined as task-related activation feature mapping. For a specific task of CP-ViT, in order to identify the important patches, we computed the gradients of the output y (before the activation function) with respect to patch features (after patch embedding) P^k, i.e. $\frac{\partial y}{\partial P^k}$. These gradients flowing back to the patch features are global-average-pooling over the feature dimensions to obtain the patch importance weights. The important weights are:

$$\alpha_k = \frac{1}{Z}\sum_{i=1}^{Z}\frac{\partial y}{\partial P_i^k} \tag{4.14}$$

where Z is the dimension of the patch embedding features. After we have the weights of all the patches, the top K patches that have the highest weights are selected and re-distributed to the core nodes. Note that the patch distribution process is not random but distributed based on the nodes' degree in a in a descending manner: the patches with higher importance weights are distributed to the core nodes with a higher degree. The algorithm for patch redistribution is detailed described in algorithm 3, and the corresponding patch redistribution process is illustrated in Fig. 4.12. As shown in Fig. 4.12, the image patches were randomly distributed at the first epoch but as the training process proceeded, patches with high gradients are identified as important patches and gradually redistributed to the core nodes. After certain iteration epochs, those patches that contribute the most to the classification result will be distributed to the core nodes.

4.2.6 Conclusion

In this work, we proactively instilled an organizational principle of BNN, that is, Core-Periphery property, to guide the design of ANN of ViT. For this, we provide a unified framework to introduce the core-periphery principle to guide the design of self-attention, the most prominent mechanism in transformers. Our extensive experiments suggest that there exist sweet spots of CP graphs that lead to CP-ViTs with significantly improved predictive performance. In general, our work advances the state of the art in three ways: 1) this work provides novel insights for brain-inspired AI by applying organizational principles of BNNs to

ANN design; 2) the optimized CP-ViT can significantly improve its predictive performance while have the potential to reduce the unnecessary computational cost; and 3) the core nodes in CP-ViT are associated with task-related meaningful image patches, which can significantly enhance the interpretability of the trained deep model.

Chapter 5

Identification of Causal Relationship between Amyloid-β Accumulation and Alzheimer's Disease Progression via Counterfactual Inference

5.1 Overview

Alzheimer's disease (AD) is a neurodegenerative disorder that is beginning with amyloidosis, followed by neuronal loss and deterioration in structure, function, and cognition. The accumulation of amyloid-β in the brain, measured through 18F-florbetapir (AV45) positron emission tomography (PET) imaging, has been widely used for early diagnosis of AD. However, the relationship between amyloid-β accumulation and AD pathophysiology remains unclear, and causal inference approaches are needed to uncover how amyloid-β levels can impact AD development. In this paper, we propose a graph varying coefficient neural network (GVCNet) for estimating the individual treatment effect with continuous treatment levels using a graph convolutional neural network. We highlight the potential of causal inference approaches, including GVCNet, for measuring the regional causal connections between amyloid-β accumulation and AD pathophysiology, which may serve as a robust tool for early diagnosis and tailored care.

5.2 Background

The differentiation of Alzheimer's disease (AD) from the prodromal stage of AD, which is the mild cognitive impairment (MCI), and normal control (NC) is an important project that interests many researchers making effort on (Q. Li et al., 2017; M. B. Miller et al., 2022). It is commonly recognized through studies that the progression of AD involves a series of gradually intensifying neuropathological occurrences. The process begins with amyloidosis, followed by neuronal loss and subsequent deterioration in the areas of structure, function, and cognition (Ossenkoppele et al., 2022). As a non-invasive method that could measure the accumulation of amyloid in the brain, 18F-florbetapir (AV45) positron emission tomography (PET) imaging has been widely used for early diagnosis of AD (Q. Ge et al., 2022). The use of florbetapir-PET imaging to characterize the deposition of amyloid-β has shown to be of significant diagnostic value in identifying the onset of clinical impairment.

In recent years, there has been increasing research in counterfactual causal inference to estimate the treatment effect in various domains such as medicine (B.-M. Lv et al., 2021; Meilia et al., 2020; Yazdani & Boerwinkle, 2015), public health (Glass et al., 2013; Glymour & Spiegelman, 2017; Rothman & Greenland, 2005), and marketing (Hair Jr & Sarstedt, 2021; Varian, 2016). Especially, estimating the causal effect of continuous treatments is crucial. For example, in precision medicine, a common question is *"What is the ideal medicine dosage to attain the best result?"*. Therefore, an average dose-response function (ADRF) that elucidates the causal relationship between the continuous treatment and the outcome becomes imperative.

Estimating the counterfactual outcome presents a significant challenge in causal effect estimation, as it is inherently unobservable. To provide a clear definition, we use the binary treatment scenario ($T = 1$ or $T = 0$) for illustration. As depicted in Fig. 5.1, let us consider a patient with a headache (x_i) who has the option to either take the medicine ($T = 1$) or not take it ($T = 0$). The potential outcomes corresponding to these two treatment choices would be being cured ($Y_i(T = 1)$) or not being cured ($Y_i(T = 0)$), respectively. The causal effect is defined as the difference between these two potential outcomes. However, given that a patient can only choose one treatment option, we can observe only one outcome (the observed outcome), while the other outcome that was not observed is considered the counterfactual outcome. Similarly, in the context of a continuous setting, estimating the counterfactual outcome remains a significant challenge.

Therefore, a variety of existing works on causal effect estimation focus on counterfactual estimation (Hassanpour & Greiner, 2019; Johansson et al., 2016; Morgan & Winship, 2015) under the assumption of binary treatments or continuous treatments (ADRF estimation) (Bica et al., 2020; Hirano & Imbens, 2004; L. Nie et al., 2021; Schwab et al., 2020; Y. Zhang et al., 2022).

Especially, in the context of continuous treatments, the generalized propensity score (GPS), proposed by Hirano and Imbens (Hirano & Imbens, 2004), is a traditional approach to estimate ADRF with counterfactual outcomes. Moreover, as machine learning has gained increasing attention due to its extraordinary ability to solve complex problems, many existing works use machine learning techniques to address the problem. Schwab et al. (Schwab et al., 2020) proposed DRNet to split a continuous treatment into several intervals and built separate prediction heads for them on the latent representation of input. Nie et al. (L. Nie et al., 2021) adopted varying coefficient structure to explicitly incorporate continuous treatments as a variable for the parameters of the model, preserving the continuity of ADRF. Other methods, such as GAN (Bica et al., 2020) and transformer (Y. Zhang et al., 2022), have also been proposed.

In this work, we propose a novel model, the Graph Varying Coefficient Neural Network (GVCNet), for measuring the regional causal associations between amyloid-β accumulation and AD pathophysiology. Specifically, by comparing our model with the most advanced model, VCNet, we demonstrate that our model achieves better performance in AD classification. Moreover, we adopt K-Means clustering to group the generated average dose-response function (ADRF) curves from each region of interest (ROI) and then map them onto the cortical surface to identify the amyloid-β positive regions.

The main contributions of this work are summarized as follows:

1. To the best of our knowledge, this is the early attempt to utilize the brain structural topology as the graph to measure the regional causal associations between amyloid-β accumulation and AD pathophysiology. Consistent experimental results on AD public dataset not only demonstrate the effectiveness and robustness of the proposed framework, but also support this hypothesis: the AD pathophysiology is deeply associated with amyloid-β accumulation, no matter with which kind of topology graph. 2. Compared with the most advanced approach (i.e., VCNet), the proposed GVCNet experimentally obtains a higher diagnosis accuracy, suggesting that the good performance could be achieved with graph topology. As such our framework, such attempt extends the applications of graph-based algorithms on brain imaging analysis and

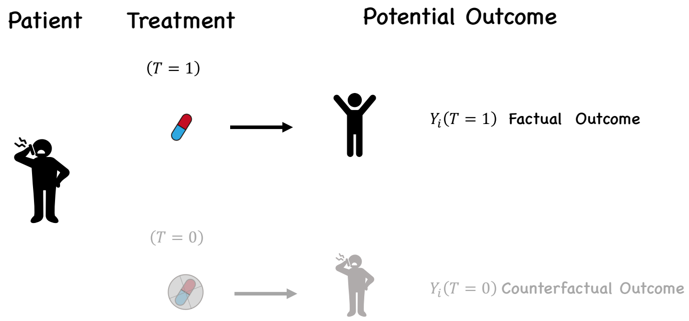


Figure 5.1: An Example of counterfactual problem: A patient with a headache who takes medicine and is cured. While the counterfactual scenario, i.e., the outcome had the patient not taken the medicine, is unobserved.

provides a new insight into the causal inference that combines the phenotype, structural and functional data. 3. Our work demonstrates clearly that there are four brain regions (i.e., pre- & post- central gyrus among cortical area, left & right pallidum among subcortical area) can be as the key ROIs for AD diagnosis. With the quantitative experimental results, with such ROIs, the diagnosis accuracy is better than with the whole brain information.

5.3 Related Work

5.3.1 Counterfactual Outcome Estimation

The definition of counterfactual outcome is typically framed using the potential outcome framework (Rubin, 1974). To provide a clear definition, we illustrate with the use of binary treatments, which can be extended to multiple treatments by comparing their potential outcomes. Each individual x_i has two potential outcomes: $Y_i(T=1)$ and $Y_i(T=0)$, corresponding to the two possible treatments ($T=1$ or $T=0$). Since an individual can only receive one of the two treatments in observational data, only one potential outcome can be observed (observed outcome), while the remaining unobserved outcome is referred to as the counterfactual outcome. Hence, the major challenge in estimating Individual Treatment Effect (ITE) lies in inferring counterfactual outcomes. Once the counterfactual outcomes are obtained, ITE can be calculated as the difference between the two potential outcomes:

$$ITE_i = Y_i(T = 1) - Y_i(T = 0). \tag{5.1}$$

Many existing approaches have been proposed to estimate the counterfactual outcomes, such as conditional outcome modeling that trains two separate models to predict outcomes for the treatment group and control group and use the predicted value to fill the unobserved counterfactual outcomes. In addition, tree-based and forest-based methods are widely used to estimate ITE (Chipman et al., 2010; Hansen, 2008; Wager & Athey, 2018). Additionally, matching methods (Morgan & Winship, 2015; Stuart, 2010), stratification mathods (L. Yao et al., 2022), deep representation methods (Hassanpour & Greiner, 2019; L. Yao et al., 2022) have been proposed to address the problem as well.

5.3.2 Continuous Treatment Effect Estimation

Continuous treatments are of great practical importance in many fields, such as precision medical. Typically, the objective of continuous treatment effect estimation is to estimate the average dose-response function (ADRF), which demonstrates the relationship between the specific continuous treatment and the outcome. Although recent works utilized the representation learning methods for ITE estimation (Chu et al., 2020; Johansson et al., 2016; Shalit et al., 2017; L. Yao et al., 2018), most of the existing works are under the assumption of binary treatments, which cannot be easily extended to continuous treatment due to their unique model design.

To address this issue, Schwab et al. (Schwab et al., 2020) extended the TARNet (Shalit et al., 2017) and proposed Dose Response networks (DRNet), which divided the continuous dosage into several equally-sized dosage stratus, and assigned one prediction head for each strata. To further achieve the continuity of ADRF, Nie et al., (L. Nie et al., 2021) proposed a varying-coefficient neural network (VCNet). Instead of the multi-head design, it used a varying coefficient prediction head whose weights are continuous functions of treatment t, which improved the previous methods by preserving a continuous ADRF and enhancing the expressiveness of the model. Hence, in this paper, we adopt it as part of the model to estimate the effect of each Regions of Interest (ROI) of the brain on Alzheimer's disease.

5.3.3 Traditional Correlation-based PET Image Analysis Methods

The correlation-based methods on PET images analysis could be used in many clinical applications, such as tumor detection and brain disorder diagnosis. An et al. used canonical correlation analysis-based scheme to estimate a standard-dose PET image from a low-dose one in order to reduce the risk of radiation exposure and preserve image quality (An et al., 2016). Landau et al. used the traditional corrlation method to compare the retention of the 11-C radiotracer Pittsburgh Compound B and that of two 18-F amyloid radiotracers (florbetapir and flutemetamol) (Landau et al., 2014). Zhu et al. used the cannoical representation to consider the correlations relationship between features of PET and other different brain neuroimage modalities (X. Zhu et al., 2014). Li et al. used sparse inverse covariance estimation to reveal the relationship between PET and structural magnetic resonance imaging (sMRI) (Q. Li et al., 2018).

And for the AD diagnosis, it has been suggested that brain regions such as the posterior cingulate and lateral temporal cortices are affected more in AD than the NC, with the florbetapir-PET (Camus et al., 2012). Some researches on florbetapir-PET imaging have revealed that neurodegeneration does not influence the level of amyloid-β accumulation. Instead, amyloid-β pathophysiology is considered a biologically independent process and may play a "catalyst" role in neurodegeneration (Jack et al., 2014). There have also been many theories that highlight the amyloid-β pathologies as the main driving forces behind disease progression and cognitive decline. In order to characterize the relationship between the amyloid-β accumulation and AD pathophysiology, the counterfactual causal inference method will be a useful tool to uncover how the patterns of causality or significant changes in regional or temporal amyloid-β levels can impact the development of AD over time.

5.3.4 Graph Neural Network

Deep learning has revolutionized many machine learning tasks, but challenges arise when data is represented as graphs. The basic idea behind GNNs is to iteratively update the feature vectors of each node by aggregating the feature vectors of its neighboring nodes.

The update rule for a GNN can be formalized as follows:

$$h_i^{l+1} = \sigma(\mathbf{a}_i^l W^l), \mathbf{a}_i^l = g^l(h_i^l, \{h_u^l : u \in \mathcal{N}(i)\}), \tag{5.2}$$

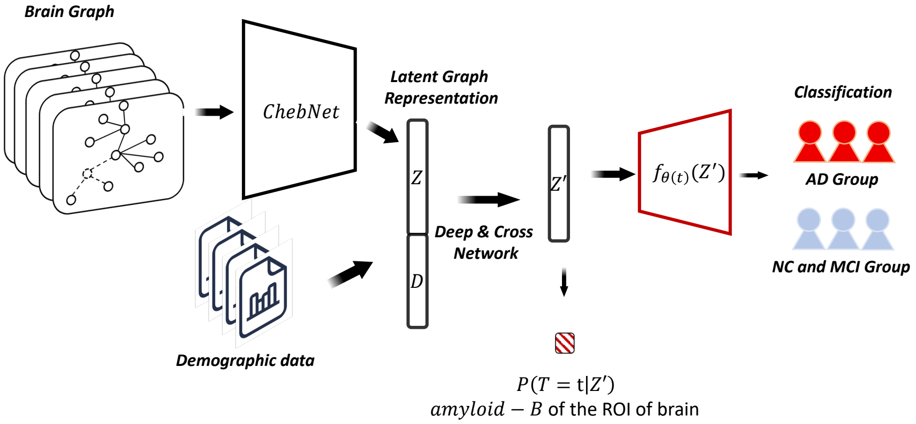


Figure 5.2: The framework of GVCNet for AD classification and individual treatment effect estimation. (a) we utilize ChebNet for feature embedding and then integrate treatment in the following dynamic fully connected layer for AD classification task. (b) We employee KMeans cluster algorithm to cluster the individual ADRFs into 3 groups: aβ-positive (up), aβ-negative (down) and aβ-neutral and mapping these groups on the brain.

where $h_i^{(l+1)}$ is the feature vector of node i at layer $l+1, \mathcal{N}(i)$ is the set of neighboring nodes of i, g^l is the aggregation function at latyer l, and $W^{(l)}$ is a learnable weight matrix at layer l. The function σ is a non-linear activation function, such as the ReLU function. Graph convolutional networks (GCNs) extend convolutional neural networks (LeCun, Bengio, et al., 1995) to the graph domain, allowing for meaningful feature extraction. GCNs have been applied in various fields, including node classification (C. Wang et al., 2017), link prediction (M. Zhang & Chen, 2018), and graph generation (Kawamoto et al., 2018). Initial work on GCNs was proposed by (Gori et al., 2005) in 2013, followed by the seminal paper by (Kipf & Welling, 2016) in 2017. Since then, many extensions and improvements to GCNs have been proposed, including Graph Attention Networks (GATs) (Veličković et al., 2017) and GraphSAGE (Hamilton et al., 2017). Researchers have also studied different graph convolutional layers, such as Message Passing Neural Networks (MPNNs) (Gilmer et al., 2017) and Convolutional Graph Neural Networks (ConvGNNs) (Schlichtkrull et al., 2018). Overall, GCNs have shown great potential in graph representation learning and have the potential to revolutionize many applications where data is represented in the form of graphs.

5.4 Methodology

5.4.1 Problem Setting

VCNet is one of the advanced methods for ADRF estimation, typically it can generate continuous ADRF and provide promising counterfactual estimation. Hence, in this study, we adopt this model to estimate the effect between the amyloid-β level and the probability of gaining AD. Typically, we treat the amyloid-β in a specific brain region as the treatment T and whether the subject gains AD as the outcome Y.

In our study, we used the Harvard-Oxford Atlas (HOA) to divide the entire brain into 69 regions. Since the some regions for tau imaging is not a target binding region, we excluded the following regions: left cerebral white matter, left cerebral cortex, left lateral ventrical, right cerebral white matter, right cerebral cortex, right lateral ventricle and brain-stem. For the rest of 62 regions, we treated one region as the treatment and used the other regions as covariates (X) to train a separate model for each setting. We iterated this process 62 times to obtain the causal effect and accuracy estimates for each region. To capture more information, we used graph structures of the whole brain denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where each graph contains 62 nodes representing 62 ROIs, $\mathcal{V}$ represents the node set and $\mathcal{E}$ represents the edge set. Let $X \in R^{N \times F}$ be the input feature matrix, where each row corresponds to a node and each column corresponds to a feature. To estimate the causal effect of one ROI, we removed the corresponding node and all edges related to it and used the rest of the graph as input (61 nodes). Finally, we used the amyloid-B value as the treatment variable T for the VCNet analysis. In our work, we follow three fundamental assumptions for identifying ADRF:

Assumption 1 ***Stable Unit Treatment Value Assumption (SUTVA)***: *There are no unit interactions, and there is only one version of each treatment, which means that various levels or doses of a specific treatment are considered as separate treatments.*

Assumption 2 ***Positivity***: *Every unit should have non-zero probability of being assigned to every treatment group. Formally,* $P(T = t|X = x) \neq 0, \forall t \in \mathcal{T}, \forall x \in X$.

Assumption 3 ***Ignorability***: *Given covariates* x, *all potential outcomes* $\{Y(T = t)\}_{t \in \mathcal{T}}$ *are independent of the treatment assignment, implying that there are no unobserved confounders. Mathematically,* $\{Y(T = t)\}_{t \in \mathcal{T}} \perp\!\!\!\perp T|X$.

Table 5.1: Data Description.

	Groups	N	Age	*p*-value	Sex(M/F)	*p*-value	MMSE Score	*p*-value	CDR Score	*p*-value
ADNI1	NC	100	75.83 4.71	0.4416	61/39	0.3923	28.94 1.12	0	0.0 0.0	0
	MCI	205	74.98 7.23		136/69		27.18 1.69		0.49 0.03	
	AD	92	75.87 7.33		54/38		23.48 2.11		0.81 0.244	
ADNI2	NC	159	76.63 6.33	0.1436	77/82	0.2099	28.63 1.69	0	0.09 0.21	0
	MCI	143	75.04 7.43		74/69		24.71 4.50		0.68 0.53	
	AD	106	76.29 7.95		63/43		20.02 4.60		1.06 0.48	

N is the number of participants in such group; *p*-value is calculated based on ANOVA; M means male; F means female

Table 5.2: Evaluation on GVCNet. $\star$ means the demographic feature is selected.

Dataset	Graph	Age	Sex	MMSE	CDR	Accuracy (%)
ADNI1+ADNI2	Corr					0.8296 ± 0.0020
ADNI1+ADNI2	Corr	$\star$	$\star$			0.8675 ± 0.0018
ADNI1+ADNI2	Corr	$\star$	$\star$	$\star$	$\star$	0.8868 ± 0.0027
ADNI1+ADNI2	DTI					0.8698 ± 0.0019
ADNI1+ADNI2	DTI	$\star$	$\star$			0.8689 ± 0.0018
ADNI1+ADNI2	DTI	$\star$	$\star$	$\star$	$\star$	0.8872 ± 0.0022

5.4.2 GVCNet

In our proposed GVCNet framework, as illustrated in Figure 7.1, there are three main components: ChebNet (Defferrard et al., 2016), Deep&Cross Network (R. Wang et al., 2017), and VCNet (L. Nie et al., 2021). These components work together to estimate the Average Treatment Effect (ATE) using graph-structured data and demographic information.

The ChebNet component takes advantage of the graph structure of the data and utilizes this graph structure to generate features or representations that capture the underlying relationships between entities.

The Deep&Cross Network component incorporates demographic data into the framework. The Deep&Cross Network module utilizes these demographic features to learn complex interactions between them, capturing both low-order and high-order feature interactions. This helps to capture additional information beyond what can be learned solely from the graph-structured data.

The resulting latent representation, denoted as Z', which is a combination of features from ChebNet and Deep&Cross Network, is then fed into the VCNet component. VCNet infers the treatment distribution from Z' to ensure that it contains sufficient information for accurate ADRF estimation. Finally, the ADRF is estimated based on t and Z'.

5.4.3 ChebNet

In this paper, to preserve the topological information of PET data. We introduce the Chebyshev neural network (ChebNet) (Defferrard et al., 2016) to replace the first two fully connected layers in VCNet. ChebNet uses Chebyshev polynomials to approximate the graph Laplacian filter, which is a commonly used filter in GCNs. Chebyshev polynomials are a sequence of orthogonal polynomials that can be used to approximate any smooth function on a given interval, and can be efficiently computed using recursive formulas.

The equation of first ChebNet is as follows:

$$f_{\text{out}}(\mathcal{L}, \mathbf{X}) = \sigma\left(\sum_{k=0}^{K-1} \Theta_k T_k(\tilde{\mathcal{L}})\mathbf{X}\right) \tag{5.3}$$

where $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the input matrix of N nodes, each with F features, $\mathcal{L}$ is the graph Laplacian, and $\tilde{\mathcal{L}}$ is the normalized Laplacian defined as $\tilde{\mathcal{L}} = 2\mathcal{L}/\lambda_{\max} - I_N$, where $\lambda_{\max}$ is the largest eigenvalue of $\mathcal{L}$. $T_k(\cdot)$ are Chebyshev polynomials of order k and Θ_k are the learnable filter coefficients for the k-th Chebyshev polynomial. Finally, $\sigma(\cdot)$ is a non-linear activation function such as ReLU or sigmoid that is applied element-wise to the output of the ChebNet. And the binary cross-entropy loss function is utilized to quantify the dissimilarity between the predicted probability of the positive class and its true probability in binary classification tasks.

5.4.4 Deep & Cross Network

The Deep & Cross Network (DCN) (R. Wang et al., 2017) is utilized to combine demographic data with topological information from PET data. Instead of conducting task-specific feature engineering, the DCN is capable of automatically learning the interactions between features that contribute to the task. Although deep neural networks (DNNs) are capable of extracting feature interactions, they generate these interactions in an implicit way, require more parameters, and may fail to learn some feature interactions efficiently.

The DCN uses an embedding and a stack layer to embed sparse features in the input into dense embedding vectors $x_{embed,k}^{T}$ to reduce the dimension. These vectors are then stacked with normalized dense features x_{dense}^{T} in the input as a single vector $x_0 = [x_{embed,1}^{T}, ..., x_{embed,k}^{T}, x_{dense}^{T}]$. A cross network and a deep network are adopted to further process this vector in parallel. The hallmark of the paper is the cross network, which applies explicit and efficient feature crossing as shown below:

$$x_{l+1} = x_0 x_l^T w_l + b_l + x_l \tag{5.4}$$

Here, x_l denotes the output of the l-th cross layer, and w_l and b_l represent the weight and bias of the l-th cross layer, respectively. The equation demonstrates that the degree of feature interactions grows with the depth of the layer. For example, the highest polynomial degree of x_0 of an l-layer cross network is $l+1$. Additionally, the interactions in the deep layer depend on the interactions in shallow layers.

In addition to the cross network, a fully-connected feed forward neural network is used to process x_0 simultaneously. The outputs of the cross network and the deep network are concatenated and fed into a standard logit layer to conduct the final prediction by the combination layer.

5.4.5 VCNet

Despite the prior endeavours on ITE estimation, most of the work are focused on binary treatment settings and fail to extend to continuous treatment easily. Although some papers propose to estimate the continuous treatment by splitting the range of treatment into severel intervals and use one prediction network for each interval, the continuity of ADRF is still an open issue. To address these issues, VCNet is proposed by (L. Nie et al., 2021), which is capable of estimating continuous treatment effect and maintaining the continuity of ADRF simultaneously.

A fully connected feedforward neural network is trained to extract latent representation z from input x. To guarantee z encode useful features, z is used to estimate the conditional density of the corresponding treatment $\mathbb{P}(t|z)$ through a conditional probability estimating head. Specifically, $\mathbb{P}(t|z)$ is estimated based on the $(B+1)$ equally divided grid points of treatment and the conditional density for the remaining t-values is computed using linear interpolation. After obtaining the z containing valuable information, a varying coefficient neural network $f_{\theta(t)}(z)$ is adopted to predict the causal effect of t on the outcome $y_{i,t}$ based on z and the corresponding t, where the network parameters are a function of treatment $f_{\theta(t)}$ instead of fixed parameters. Typically, the B-spline is used to model $\theta(t)$:

$$\theta(t) = [\sum_{l=1}^{L} a_{1,l}\varphi_l^{\text{NN}}(t), ..., \sum_{l=1}^{L} a_{d_{\theta(t)},l}\varphi_l^{\text{NN}}(t)]^T \in \mathbb{R}^{d(\theta)}, \tag{5.5}$$

$\varphi_l^{\mathrm{NN}}(t)$ denotes the spline basis of the treatment and $a_{1,l}$ are the coefficients to be learned; $d(\theta)$ is the dimension of $\theta(t)$. By utilizing the varying coefficient neural network, the influence of the treatment effect t on the outcome is integrated via the parameters of the outcome prediction network, thereby preventing any loss of treatment information. Additionally, the incorporation of t in this manner allows for the attainment of a continuous ADRF.

5.5 Experiment

5.5.1 Dataset

In this paper, we conducted an evaluation of their proposed algorithm using two subsets of data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu), specifically ADNI-1 and ADNI-2, as well as the entire dataset. The subjects were divided into three categories, consisting of AD, NC, and MCI, as shown in Table 5.1. In this paper, we take AD as the AD group (298 subjects) and NC+MCI as the non-AD group (607 subjects). All florbetapir-PET images were co-registered with each individual's sMRI and subsequently warped to the cohort-specific DARTEL template. And all subject has demographic features: age, sex, CDR score and MMSE score.

All sMRI and florbetapir-PET images in this study are pre-processed by FMRIB Software Library (FSL) 6.0.3 (https://fsl.fmrib.ox.ac.uk/). The brain extraction step is based on the BET algorithm firstly(Smith, 2002). And the skull is stripped from the source image sapce. Secondly, the sMRI images are aligned to Montreal Neurological Institute T1 standard template space (MNI152) with the FLIRT linear registration algorithm(Jenkinson et al., 2002), which can save computational time during the application stage. All florbetapir-PET images were co-registered with each individual's sMRI and subsequently warped to the cohort-specific DARTEL template. More specifically, after registration, the sMRI and florbetapir-PET images are cropped to the size of 152 × 188 × 152 by removing the voxels of zero values in the periphery of brain. Then, all the images are downsampled to the size of 76 × 94 × 76 that to reduce the computational complexity. And all subject has demographic features: age, sex, CDR score and MMSE score.

In order to generate the structural connectivity matrix between different cortical regions, we also used the T1w and diffusion MRI (dMRI) provided in the ADNI database. T1-weighted images were acquired using a 3D sagittal MPRAGE volumetric sequence with TE = 3.0 ms; TI = 900.0 ms; TR = 2300.0 ms; flip angle = 9°; matrix size = 176 × 240 × 256; voxel size = 1.2 × 1.1 × 1.1

mm3. dMRI was acquired with a spin-echo planar imaging (EPI) sequence. 48 noncollinear gradient directions were acquired with a b-value of 1,000 s/mm2. 7 additional volumes were acquired without diffusion weighting (b-value = 0 s/mm2). Other parameters of dMRI were as follows: TE = 56.0 ms; TR = 7200.0 ms; flip angle = 90°; matrix size = 116 × 116 × 80; isotropic voxel size = 2 × 2 × 2 mm3. A subset of 20 subjects was used for generating a group-wise connectivity matrix. For each subject, whole brain tractography was computed using the dMRI data, with the Unscented Kalman Filter (UKF) tractography method (Wan & Van Der Merwe, 2000, 2001) provided in the SlicerDMRI (Norton et al., 2017; F. Zhang et al., 2020) software. Structural T1w imaging data was processed using FreeSurfer (version 6.0, https://surfer.nmr.mgh.harvard.edu/), and cortical regions were parcellated with the Desikan-Killiany Atlas (Alexander et al., 2019). Co-registration between the T1-weighted and dMRI data was performed using FSL (Jenkinson et al., 2012). Then, for each pair of cortical regions, streamlines that end in the two regions were extracted and the number of streamlines were computed, followed by the creation of the subject-specific connectivity matrix. For the group-wise connectivity matrix, the mean number of streamlines across the 20 subjects was recorded.

In the trainning process, We randomly split the dataset into a training set (633 subjects) and a testing set (272 subjects). The proposed model was tested on the testing set to calculate the classification accuracy and generate average dose-response function curves (ADRFs) for each ROI.

5.5.2 Experiment Setting

In GVCNet, we designate each one of the 62 ROIs as the treatment and use the other ROIs as patient features. The average amyloid-β level serves as the signal for each ROI. We construct the input graph by defining the ROIs as nodes V and the DTI structure among the ROIs as edges E. For the sturctural connectivity matrix, we have two alternative cunstructing options as follows: one is to use the Pearson correlation value among the ROIs' T1-weighted values to construct the structural correlation graph (which is called the Corr graph in this paper to make it simplified); the other is to use the smoothed white fibers among the ROIs based on the 20 subjects (which is called DTI graph). Then treat the graph embedding and demographic data as input of the deep and cross network. Finally, feed the treatment and calculate the counter-factor with our GVCNet. For the hyper-parameters, we set the learning rate to 1e-4 and β to 0.5. During model training, all networks were trained for 600 epochs. Our model is trained using Adam (Kingma & Ba, 2014) with momentum as 0.9.

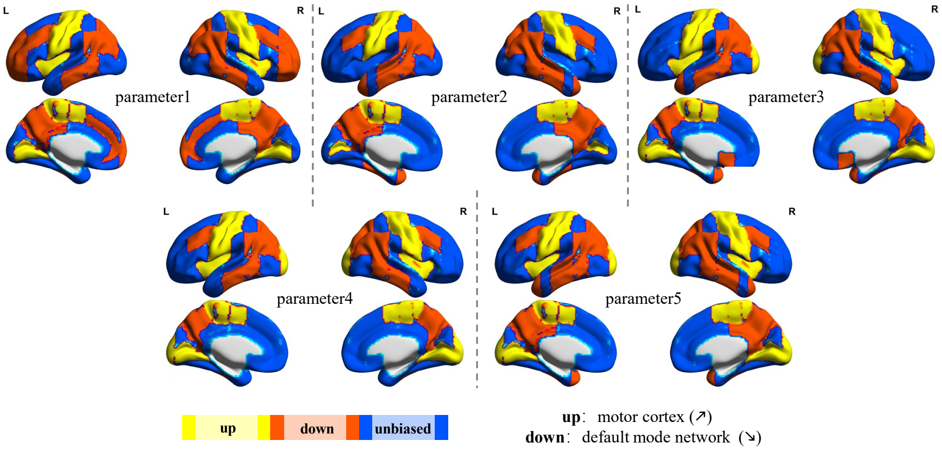


Figure 5.3: The cortical curve trends clustered by k-means.

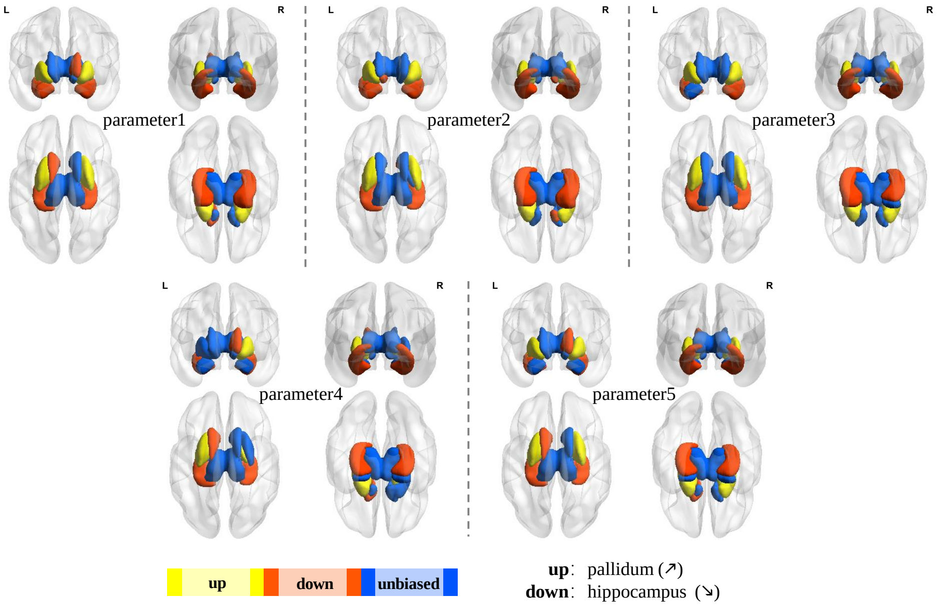


Figure 5.4: The subcortical curve trends clustered by k-means.

Table 5.3: Evaluation on GVCNet and VCNet on ADNI1+ADNI2

	Average Accuracy
VCNet	0.8401 ± 0.0048
Graph-VCNet	**0.8872 ± 0.0022**

5.5.3 Prediction Performance

First, we compare our model, GVCNet with the baselien model, VCNet. As shown in Table 5.3, the prediction performance of our model is around 88.72%, which is 4.7% higher than VCNet. In Table 5.2, we evaluate the model's performance by the accuracy percentage. The table presents the evaluation results of the GVCNet model on different datasets, using different types of graphs, and considering different demographic factors.

The first three rows present the evaluation results on the combined ADNI1+ADNI2 dataset, using Corr graphs and again different combinations of demographic factors. The model achieves an average accuracy of 0.8296 when no demographic features are selected, an average accuracy of 0.8675 when age and sex are used, and an average accuracy of 0.8868 when all the demographic features are selected.

The last three rows present the evaluation results on the combined ADNI1+ADNI2 dataset, using DTI graphs and again different combinations of demographic factors. The model achieves an accuracy of 0.8698 when no features are selected, an accuracy of 0.8689 when age and sex features are considered, and an accuracy of 0.8872 when all the features are selected. By comparing the last 6 rows, we can see that using DTI as the graph structure is slightly better than using the correlation graph between the ROIs as the graph structure.

5.5.4 ADRF Curve Analysis

Based on the patterns of the estimated ADRF of each region and the premise that different parts of the brain may play different roles during the normal/abnormal aging process, we use KMeans clustering method to cluster the ADRF curves from each region into three groups: upward(up, aβ positively respond to the treatment), downward(down, aβ negatively respond to the treatment) and unbiased, based on their trend of relationship with AD probability. Brain regions within each cluster were visualized onto the cortex and subcortex mappings in Fig. 5.3 and Fig. 5.4. It can be found that there exist strong causal relationships between the AD progression and the PET signal level in the precentral/postcentral gyrus (cortical) and left/right pallidum (subcortical), indicating the potentially important role of these regions in modulating the Amyloid-β protein pathway in AD. It is interesting to observe that both the cortical (precentral gyrus) and subcortical (pallidum) regions responsible for voluntary motor movements (Banker & Tadi, 2019; Freund, 2002) are all highly responding to AD, indicating a possible link between the behavior and pathological aspect of AD.

In addition, based on Table 5.4 that brain regions in the up group will have a slightly higher prediction power towards the AD probability, we investigated the patterns of ADRF curves and the regions within the up group in Fig. 5.5, which is consistent with Figs. 5.3 and 5.4 that pre- and post- central gyrus, left and right pallidum are upward with the increasing treatment. Moreover, we can obtain the same conclusion from both the VCNet and GVCNet, as shown in Fig. 5.6. Compared with the VCNet, our proposed Graph-VCnet can achieve much better prediction accuracy no matter with which kind of brain regions. And more specifically, with upward brain regions, both VCNet and Graph-VCNet could achieve the best prediction accuracy, compared with the other kinds of brain regions.

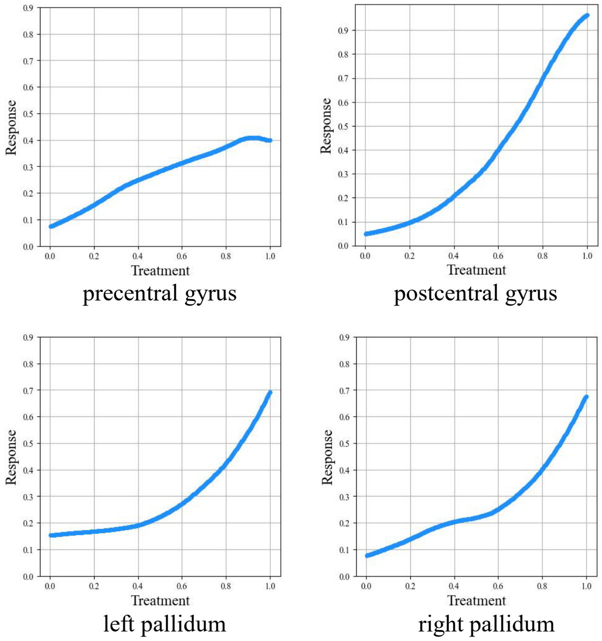
Table 5.4: KMeans Cluster Accuracy

Cluster	Accuracy
Down	0.8836 ± 0.0034
Unbiased	0.8822 ± 0.0035
Up	**0.8915 ± 0.0018**

5.6 Conclusion and Discussion

In this chapter, we propose a novel model called GVCNet, which combines a graph neural network architecture with a targeted regularization approach to estimate varying coefficients of a treatment effect model and improve the model's performance. Experiment results show that GVCNet exhibits promising capabilities in making counterfactual causal inferences for Alzheimer's Disease (AD) progression based on the regional level of Amyloid-beta protein.

The rationalization for employing a graph neural network architecture in GVCNet stems from the inherent complexity and interconnectedness of brain



precentral gyrus

postcentral gyrus

left pallidum

right pallidum

Figure 5.5: ADRF for the typical upward ROIs.

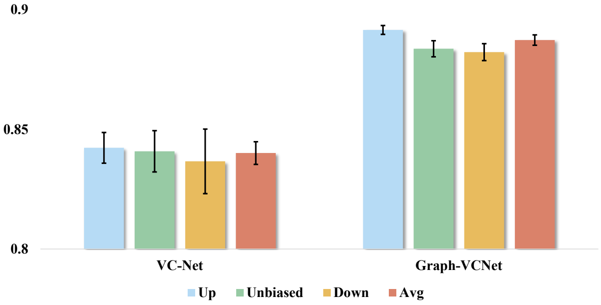


Figure 5.6: Prediction accuracy with VCNet and Graph-VCNet based on different brain regions.

regions, both structurally, functionally, and pathologically. The graph structure allows for capturing the potentially long-distance spatial relationships and dependencies among these regions, providing a more comprehensive representation of the underlying proteinopathy dynamics. Furthermore, GVCNet incorporates a targeted regularization approach. Regularization techniques play a crucial role in mitigating model complexity and ensuring robustness. By imposing the proposed regularization constraints, GVCNet can effectively handle the inherent noise and variability in PET imaging data, leading to more reliable, generalizable, and accurate predictions.

The potential of GVCNet in patient management, treatment, and drug discovery is substantial. If the model demonstrates sufficient robustness and consistency through rigorous validation studies, it can be ultimately utilized to project personalized AD progression trajectories. By leveraging counterfactual analysis, GVCNet can provide insights into the "what if" scenarios by assessing how the current imaging results would evolve if they were to worsen (due to disease progression) or improve (because of the medications or other types of interventions). This information is invaluable in guiding clinicians and patients in making informed decisions about treatment strategies and long-term care plans. Moreover, GVCNet's ability to predict the personalized treatment effect of a patient after administering a medication targeting Amyloid-beta deposition is of significant clinical importance. It can provide insights into the expected outcomes and help determine the optimal dosage for individual patients. This personalized, regional treatment prediction can aid in tailoring interventions and optimizing therapeutic strategies, leading to improved patient outcomes and more efficient use of resources.

Looking ahead, the future of imaging-guided diagnosis, prognosis, and treatment planning for AD is likely to focus on unraveling the underlying mechanisms that link imaging targets, such as Amyloid-beta protein, with the patient's internal and external characteristics (e.g., genetic factors, health conditions, comorbidities, and social determinants of health) to the disease progression. The proposed counterfactual causal inference modeling approach with multi-modal data input, as demonstrated by GVCNet, will play a pivotal role in this pursuit. With more data modalities and holistic patient characterization, we can uncover critical insights into the disease's pathophysiology, identify novel therapeutic targets, and develop more effective interventions.

In conclusion, counterfactual causal inference modeling such as GVCNet holds immense potential for advancing our understanding of personalized AD management. It will enable personalized projections of disease trajectories and treatment effects, empowering clinicians and patients to make informed decisions. The integration of imaging-guided diagnosis, prognosis, and mechanistic insights will shape the future of AD research and pave the way for improved patient care and therapeutic strategies.

Chapter 6

AugGPT: Leveraging ChatGPT for Text Data Augmentation

6.1 Overview

Text data augmentation is an effective strategy for overcoming the challenge of limited sample sizes in many natural language processing (NLP) tasks. This challenge is especially prominent in the few-shot learning scenario, where the data in the target domain is generally much scarcer and of lowered quality. A natural and widely-used strategy to mitigate such challenges is to perform data augmentation to better capture the data invariance and increase the sample size. However, current text data augmentation methods either can't ensure the correct labeling of the generated data (lacking faithfulness) or can't ensure sufficient diversity in the generated data (lacking compactness), or both. Inspired by the recent success of large language models, especially the development of ChatGPT, which demonstrated improved language comprehension abilities, in this work, we propose a text data augmentation approach based on ChatGPT (named AugGPT). AugGPT rephrases each sentence in the training samples into multiple conceptually similar but semantically different samples. The augmented samples can then be used in downstream model training. Experiment results on few-shot learning text classification tasks show the superior performance of the proposed AugGPT approach over state-of-the-art text data augmentation methods in terms of testing accuracy and distribution of the augmented samples. Codes of AugGPT are available at https://github.com/yhydhx/AugGPT.

6.2 Background

The effectiveness of natural language processing (NLP) heavily relies on the quality and quantity of the training data. With limited training data available, which is a common issue in practice due to privacy concerns or the cost of annotations, it can be challenging to train an accurate NLP model that generalizes well to unseen samples. The challenge of training data insufficiency is especially prominent in few-shot learning (FSL) scenarios, where the model trained on the original (source) domain data is expected to generalize from only a few examples in the new (target) domain (Y. Wang et al., 2020). Many FSL methods have shown promising results in overcoming this challenge in various tasks.

Existing FSL methods mainly focus on improving the learning and generalization capability of the model via better architectural design (C. Wang et al., 2021; Yin, 2020), leveraging pre-trained language models as the basis and then fine-tuning it using limited samples (Devlin et al., 2018) with meta-learning (Lee et al., 2022; Yin, 2020) or prompt-based methods (Brown et al., 2020; Han et al., 2022; Lester et al., 2021; J. Wang et al., 2022). However, the performance of these methods is still intrinsically limited by the data quality and quantity in both the source and target domains.

Besides model development, text data augmentation can also overcome the sample size limit and work together with other FSL methods in NLP (Kumar et al., 2019; Wei & Zou, 2019b). Data augmentation is usually model-agnostic and involves no change to the underlying model architecture, which makes this approach particularly practical and applicable to a wide range of tasks. In NLP, there are several types of data augmentation methods. Traditional text-level data augmentation methods rely on direct operations on the existing sample base. Some frequently used techniques include synonym replacement, random deletion, and random insertion (Feng et al., 2021). More recent methods utilize language models to generate reliable samples for more effective data augmentation, including back-translation (Sennrich et al., 2015) and word vector interpolation in the latent space (Jindal et al., 2020). However, existing data augmentation methods are limited in the accuracy and diversity of the generated text data, and human annotation is still mandatory in many application scenarios (Bayer et al., 2022; Feng et al., 2021; Shorten et al., 2021).

The advent of (very) large language models (LLMs) such as the GPT family (Brown et al., 2020; Min et al., 2021) brings new opportunities for generating text samples that resemble human-labeled data (C. Zhou et al., 2023), which significantly alleviates the burden of human annotators (Z. Liu et al., 2022). LLMs are trained in self-supervised manners, which scale up with the amount of text corpus available in the open domains. The large parameter space of LLMs also allows them to store a large amount of knowledge, while large-scale pre-training (e.g., the autoregressive objective in training GPTs) enables LLMs to encode rich factual knowledge for language generation even in very specific domains (S. Wang et al., 2023). Furthermore, the training of ChatGPT follows that of Instruct-GPT (Ouyang, Wu, Jiang, Almeida, Wainwright, Mishkin, Zhang, Agarwal, Slama, Gray, et al., 2022), which utilizes reinforcement learning with human feedback (RLHF), thus enabling it to produce more informative and impartial responses to input.

Inspired by the success of language models in text generation, we propose a new data augmentation method named AugGPT, which leverages ChatGPT to generate auxiliary samples for few-shot text classification. We test the performance of AugGPT via experiments on both general domain and medical domain datasets. Performance comparison of the proposed AugGPT approach with existing data augmentation methods shows double-digit improvements in sentence classification accuracy. Further investigation into the faithfulness and compactness of the generated text samples reveals that AugGPT can generate more diversified augmented samples while simultaneously maintaining their accuracy (i.e., semantic similarity to the original labels). We envision that the development of LLMs will lead to human-level annotation performance, thus revolutionizing the field of few-shot learning and other tasks in NLP.

6.3 Related Works

6.3.1 Data Augmentation

Data augmentation, the artificial generation of new text through transformations, is widely used to improve model training in text classification. In NLP, existing data augmentation methods work at different granularity levels: characters, words, sentences, and documents.

Data augmentation at the character level refers to the randomly inserting, exchanging, replacing, or deleting of characters in the text (Belinkov & Bisk, 2017), which improves the robustness of the NLP model against noises. Another method called optical character recognition (OCR) data augmentation generates new text by simulating the errors that occur when using OCR tools to recognize text from pictures. Spelling augmentation (Coulombe, 2018) deliberately misspells some frequently misspelled words. Keyboard augmentation (Belinkov & Bisk, 2017) simulates random typo errors by replacing a selected key with another key close to it on the QWERTY layout keyboard.

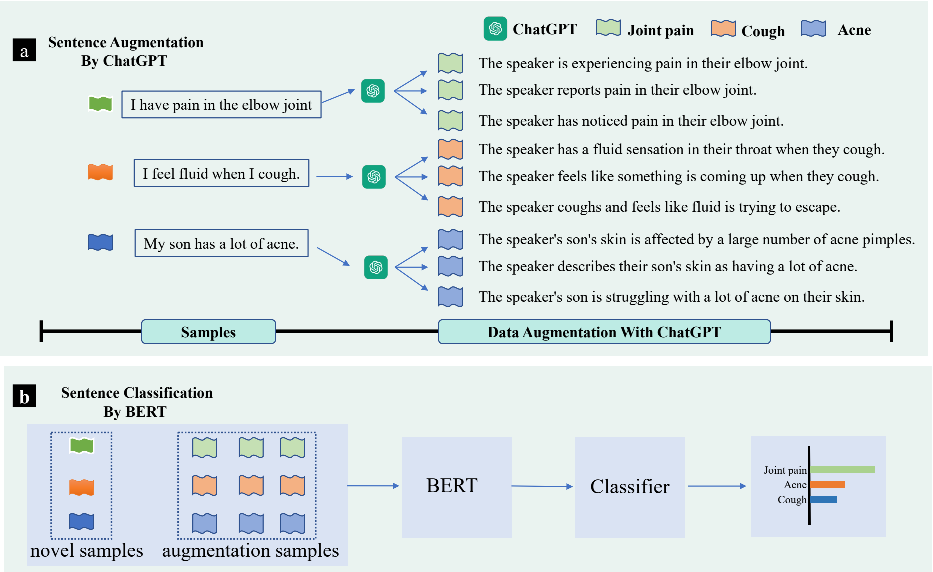

a
Sentence Augmentation By ChatGPT
ChatGPT
Joint pain
Cough
Acne
I have pain in the elbow joint
I feel fluid when I cough.
My son has a lot of acne.
The speaker is experiencing pain in their elbow joint.
The speaker reports pain in their elbow joint.
The speaker has noticed pain in their elbow joint.
The speaker has a fluid sensation in their throat when they cough.
The speaker feels like something is coming up when they cough.
The speaker coughs and feels like fluid is trying to escape.
The speaker's son's skin is affected by a large number of acne pimples.
The speaker describes their son's skin as having a lot of acne.
The speaker's son is struggling with a lot of acne on their skin.
Samples
Data Augmentation With ChatGPT
b
Sentence Classification By BERT
novel samples
augmentation samples
BERT
Classifier
Joint pain
Acne
Cough

Figure 6.1: The framework of AugGPT. a (top panel): First, we apply ChatGPT for data augmentation. We input samples of all classes into ChatGPT and prompt ChatGPT to generate samples that preserves semantic consistency with existing labelled instance. b (bottom panel): In the next step, we train a BERT-based sentence classifier on the few-shot samples and the generated data samples and evaluate the model's classification performance.

Data augmentation also works at the word level. Random swap augmentation randomly exchanges two words in the text, and random deletion augmentation randomly deletes some words (Wei & Zou, 2019a). Synonym augmentation uses synonym databases such as PPDB (Pavlick et al., 2015) to replace randomly selected words (Niu & Bansal, 2018). WordNet (G. A. Miller, 1995) is also widely used as a reference for synonym augmentation. These methods maintain semantic consistency and are suitable for text classification tasks. Wang et al. (W. Y. Wang & Yang, 2015) proposed a data augmentation method based on word embeddings, replacing words with their top-n similar words to create a new sentence. Different pre-trained word embeddings are considered (e.g., GoogleNews Lexical Embeddings (Mikolov et al., 2013)). This method is based on the principle that words close to each other in the embedding space often appear in similar contexts, which might help with maintaining grammatical consistency.

However, a serious limitation of word embedding-based methods is that close words in the embedding space are not necessarily semantically similar, yet semantic changes can affect the classification results. For example, "hot"

Table 6.1: Generated Samples From ChatGPT on PubMed20K

Original Text	Altered Text
The follow-up rate after 5 years was 85%.	1. The study showed a follow-up rate of 85% after 5 years. 2. The 5-year follow-up rate was found to be significantly high at 85%. 3. The participants had an impressive follow-up rate of 85% after 5 years. 4. The data indicated a follow-up rate of 85% 5 years post-study. 5. The 5-year follow-up rate was recorded as 85%, according to the findings. 6. The results of the study showed a follow-up rate of 85% after a 5-year period.
This means that the maturation of calluses was 27% faster in the LIPUS group.	1. The LIPUS group showed a 27% acceleration in callus maturation. 2. The results revealed that the callus maturation process was 27% faster in the LIPUS group. 3. The study found that the callus maturation rate in the LIPUS group was 27% quicker. 4. Callus maturation was 27% more rapid in the LIPUS group compared to others. 5. The callus maturation process in the LIPUS group was 27% more efficient, according to the results.
Test Registration: IRCT.ir IRCT2012071010230N1.	1. The trial was registered with the IRCT under the identifier IRCT2012071010230N1. 2. The study was officially registered with IRCT.ir, with the registration number IRCT2012071010230N1. 3. The trial was documented with IRCT.ir with the identification code IRCT2012071010230N1. 4. IRCT.ir served as the official registrar for the trial, with the registration number IRCT2012071010230N1. 5. The study was recorded with IRCT under the registration number IRCT2012071010230N1. 6. IRCT.ir recorded the trial with the identifier IRCT2012071010230N1.
While behavioral and technological interventions can lead to some modest improvements in glycemic control, these interventions have not performed much better than conventional prevention in achieving glycemic control.	1. The study found that although behavioral and technological interventions led to some slight improvements in glycemic control, they were not significantly more effective than typical care. 2. Despite the modest improvement in glycemic control through behavioral and technological interventions, they did not perform better than the standard care. 3. The results showed that while behavioral and technological interventions resulted in some minimal gains in glycemic control, they did not surpass the usual care in achieving glycemic control. 4. Although behavioral and technological interventions showed some improvement in glycemic control, they were not found to be significantly superior to the usual care. 5. The study showed that the usual care was not outperformed by behavioral and technological interventions in terms of achieving glycemic control, despite some small improvements.

and "cold" usually appear in similar contexts, so their word embeddings are close, but they have exactly opposite semantic meanings. The counter-fitting embedding augmentation (Alzantot et al., 2018; Mrkšić et al., 2016) solves this problem by using a synonym dictionary and an antonym dictionary to adjust the initial word embeddings. Specifically, the distance between embeddings of synonyms will be shortened, and the distance between embeddings of antonyms will become enlarged.

Contextual augmentation (Kobayashi, 2018; Kumar et al., 2020) is another word-level data augmentation method, which uses masked language models (MLMs) such as BERT(Devlin et al., 2019a; Sun et al., 2020), DistilBERT(Sanh et al., 2019) and RoBERTA(Y. Liu et al., 2019) to generate new text based on the context. Specifically, they insert $< mask >$ tokens in some positions of the text, or replace some words in the text with $< mask >$ tokens, and then let the MLM predict what words should be put in these masked positions. Since MLMs are pre-trained on a large number of texts, contextual augmentation can usually generate meaningful new texts.

Some text data augmentation methods work at the sentence and document level. For example, back translation (Sennrich et al., 2016) uses translation models for data augmentation. Specifically, the language model first translates the text into another language and then translates it back to the original language. Due to the randomness of the translation process, the augmented text is different from the original text, but semantic consistency is maintained. At the document level, Gangal et al. (Gangal et al., 2022) proposed a method to paraphrase the entire document to preserve document-level consistency.

In general, regardless of the granularity level or the text generation backbone (i.e., rule-based or language models), the goal of data augmentation is to produce sensible and diverse new samples that maintain semantic consistency.

6.3.2 Few-shot Learning

Deep learning has achieved remarkable success in various data-intensive applications. However, the performance of deep models could be affected if the dataset size is small in the downstream tasks. Few-shot Learning is a branch of science that focuses on developing solutions to address the challenge of small sample sizes (Fei-Fei et al., 2006; Y. Wang et al., 2020). FSL research aims to leverage prior knowledge to rapidly generalize to new tasks that contain only a few labeled samples. A classic application scenario for few-shot learning is when obtaining supervised examples is difficult or not possible due to privacy, safety, or ethical considerations. The development of few-shot learning enables practitioners to improve the efficiency and accuracy of text classification in various scenarios and deploy practical applications.

Recent advances in few-shot learning have shown promising results in overcoming the challenges of limited training data for text classification. For example, a common approach in NLP is to use a pre-trained language model such as BERT (Devlin et al., 2018) as a starting point and then fine-tune it with limited samples. Some of the most recent methodological developments (Y. Ge et al., 2022; Yin, 2020) approaches that have gained traction include prompt-tuning (Brown et al., 2020; Han et al., 2022; Lester et al., 2021; J. Wang et al., 2022) and meta-learning (Lee et al., 2022; Yin, 2020). In general, existing FSL methods target either architectural design (C. Wang et al., 2021; Yin, 2020), data augmentation (Kumar et al., 2019; Wei & Zou, 2019b) or the training process (Wei et al., 2021).

Despite the recent development of prompt-tuning and meta-learning methods, they suffer from some major limitations. For example, prompt engineering is a cumbersome art that requires extensive experience and manual trial-and-errors (Gao et al., 2021). Meta-learning, on the other hand, suffers from problems such as training instability (Antoniou et al., 2018; Finn et al., 2017; X. Yao et al., 2021) and sensitivity to hyper-parameters (Antoniou et al., 2018; Finn et al., 2017). In addition, all these FSL pipelines demand deep machine learning expertise and acquaintance with complex model architectures and training strategies, which are not attainable by common practitioners and general developers. As discussed in section 6.3.1, data augmentation is an effective solution for FSL and can be combined with other FSL models. Thus, the AugGPT method proposed in this paper, which has demonstrated the capability to generate accurate and comprehensive training samples, can overcome the issues of current FSL methods and potentially change the landscape of few-shot learning in NLP.

6.3.3 Very Large Language Models

Pre-trained language models (PLMs) based on the transformer architecture, such as the BERT (Devlin et al., 2018) and GPT (Radford et al., 2018) model families, have revolutionized natural language processing. Compared to previous methods, they deliver state-of-the-art performance on a wide range of downstream tasks and contribute to the rising popularity and democratization of language models. In general, there are three classes of pre-trained language models: autoregressive language models (e.g., the decoder-based GPT), masked language models (e.g., the encoder-based BERT), and encoder-decoder models(e.g., BART (Lewis et al., 2019) and T5 (Raffel et al., 2020)). These models typically contain between 100M and 1B parameters (Min et al., 2021).

In recent years, NLP communities have witnessed the rise of very large language models such as GPT-3 (175B parameters) (Brown et al., 2020), PaLM (540B parameters) (Chowdhery et al., 2022), Bloom (176B parameters) (Scao et al., 2022), OPT (up to 175B parameters) (S. Zhang et al., 2022), and the FLAN series (FLAN has 137B parameters) (Longpre et al., 2023). At their core, these large language models are transformer models inspired by BERT and GPT, albeit at a much larger scale.

Large language models aim to learn accurate latent feature representations of input text. These representations are often context-dependent and domain-dependent. For example, the vector representation of the word "treat" might be vastly different between medical domains and the general domain. For smaller pre-trained language models, it is often necessary to continuously pre-train and fine-tune such models to attain acceptable performance (Y. Gu et al., 2021). However, very large language models can potentially eliminate the need for fine-tuning while maintaining competitive performance (Brown et al., 2020; Rezayi, Dai, et al., 2022; C. Zhou et al., 2023).

Existing studies indicate that pre-trained language models can help augment a dataset with new samples with similar semantic meaning (Bayer et al., 2022; Feng et al., 2021), which is of significant practical value to real-world applications. In this study, we aim to use ChatGPT, a popular LLM to conduct data augmentation. ChatGPT is based on GPT-3 (Brown et al., 2020), which was trained on massive web data with diverse and rich information. Furthermore, ChatGPT was trained through Reinforcement learning from Human Feedback (RLHF). During RLHF, human feedback is incorporated into the process of generating and selecting the best results. More specifically, a reward model is trained based on human annotators' ranking or generated results. In turn, this reward model rewards model outputs that are most aligned with human preference and human values. We believe these innovations make ChatGPT the best candidate for generating human-level quality data samples.

6.3.4 ChatGPT: Present and Future

ChatGPT is a game changer in natural language processing. For the first time in human history, the power of large language models is accessible to the general public through a user-friendly chatbot interface. In turn, this common accessibility contributes to ChatGPT's unprecedented popularity. ChatGPT has emerged as a general-purpose problem solver for many NLP applications (Qin et al., 2023). Qin et al. (Qin et al., 2023) evaluated ChatGPT on a comprehensive set of NLP tasks, including common benchmarks in natural language inference, arithmetic reasoning, named entity recognition, sentiment analysis, question answering, dialogue and summarization. They conclude that ChatGPT excels in most tasks, except for tasks that focus on specific details (e.g., sequence tagging).

ChatGPT is also a valuable solution for multilingual tasks. A recent empirical study (Jiao et al., 2023) reports that ChatGPT excels at tasks involving high-resource languages (various European languages and Chinese) and is comparable with Google Translate, DeepL Translate and Tencent TranSmart. Nonetheless, ChatGPT performs poorly on low-resource languages and faces extra challenges handling distant language translation (i.e., English-German translation is considered to be less "distant", compared to English-Hindi translation). A later study (Bang et al., 2023) confirms that ChatGPT struggles with low-resource languages, although the authors observe that ChatGPT does better in understanding non-Latin scripts than generating them.

In addition, it is also possible to use the purely text-based ChatGPT to interact with multi-modal data. A group of researchers (Bang et al., 2023) use HTML Canvas and Python Turtle graphics as media for text-to-image generation. ChatGPT can faithfully generate HTML and Python code, which can be then used to generate desired images. The authors designed a flag drawing task that required ChatGPT to generate code that can generate country flags. It was found that ChatGPT could generate better flags when the prompt for code was preceded by a prompt that queries ChatGPT for the flag's description. In other words, descriptive text prompts could improve multimodal task performance.

Beyond computer science, ChatGPT can be readily applied to medical report generation and comprehension (Antaki et al., 2023; Shen et al., 2023), education (Baidoo-Anu & Owusu Ansah, 2023; Kung et al., 2023; Pavlik, 2023), rigorous math research (Frieder et al., 2023) and finance (Dowling & Lucey, 2023). Overall, ChatGPT is a versatile tool that promotes general AI usage.

However, researchers are also cautious about the possible negative impact of ChatGPT. Some of the more prominent concerns are related to bias (McGee, 2023; van Dis et al., 2023), ethics (Blum, 2022; Jabotinsky & Sarel, 2022), plagiarism (Khalil & Er, 2023; Susnjak, 2022) and job replacement *en masse* (Castelvecchi, 2022; Zarifhonarvar, 2023). In response, a commentary published in Nature advocates for urgent attention to accountability, open-source large language models and societal embrace of AI (van Dis et al., 2023).

6.4 Dataset

We first use an open domain dataset Amazon to verify the effectiveness of our method. Then, we use clinical natural language processing (clinical NLP) as the task and carry out our experiments on two popular public benchmarks. Data augmentation is particularly in demand in clinical NLP, because the significant burden of expert annotation and stringent privacy regulations make large-scale data labeling infeasible. We will describe these datasets in detail in the following sections.

6.4.1 Amazon dataset

Amazon(Bao et al., 2019; R. He & McAuley, 2016; S. Wang, Liu, et al., 2022) contains customer reviews from 24 product categories. The task is to classify reviews into their respective product categories. Since the original Amazon product dataset is proverbially large, we sample a subset of 300 samples from each category.

6.4.2 Symptoms Dataset

This dataset is published on Kaggle[1]. It contains the audio data of common medical symptom descriptions over 8 hours. We use the text transcripts corresponding to the audio data and perform sample de-duplication, and use them as model input. The dataset after preprocessing includes 231 samples of 7 symptom categories. Every example represents a sentence describing the provided

[1] https://www.kaggle.com/datasets/paultimothymooney/medical-speech-transcription-and-intent

symptoms, and the task is to classify the sentence into the corresponding symptoms.

6.4.3 PubMed20k Dataset

The PubMed20K dataset is an extensively utilized resource in NLP and text mining research, comprising around 20,000 annotated scientific abstracts from the biomedical field. These annotations encompass named entities, relationships between entities, and various semantic roles, making the dataset valuable for diverse NLP tasks such as named entity recognition, relation extraction, and text classification. The dataset originates from the PubMed database, which spans a wide array of biomedical subjects. Owing to its substantial size, variety, and high-quality annotations, PubMed20K has emerged as a popular benchmark dataset for assessing the performance of machine learning models in the realm of biomedical NLP. The abstracts in the PubMed 20K dataset undergo preprocessing and segmentation into individual sentences. Each sentence is labeled with one of the following five categories: background, objective, method, result, or conclusion. The task is to map the input sentences to their corresponding categories.

6.5 Method

6.5.1 Overall Framework

Given a base dataset $D_b = \{(x_i, y_i)\}_{i=1}^{N_b}$ with a label space $y_i \in Y_b$, a novel dataset $D_n = \{(x_j, y_j)\}_{j=1}^{N_n}$ with a label space $y_j \in Y_n$, and $Y_b \cap Y_n = \emptyset$. In the few-shot classification scenario, the base dataset D_b has a relatively larger set of labeled samples, while the novel dataset D_n has only a few labeled samples. The performance of few-shot learning is evaluated on the novel dataset. Our goal is to train a model with both base and limited novel datasets, while achieving satisfying generalizability on the novel dataset.

The overall framework of AugGPT is shown in Fig 6.1, and the training steps are shown in Algorithm 4. First of all, we fine-tune BERT on D_b. Then, the D_n^{aug} is generated by data augmentation with ChatGPT. Finally, we fine-tune BERT with D_n^{aug}.

6.5.2 Data Augmentation with ChatGPT

Similar to GPT (Radford et al., 2018), GPT-2 (Radford et al., 2019), and GPT-3 (Brown et al., 2020), ChatGPT belongs to the family of autoregressive language

Algorithm 4 The framework of AugGPT for few-shot text classification.

```
Input: base dataset $D_b$ and novel dataset $D_n$
Initialize: Initialized pre-trained BERT $model$
Definition: $D^{'}$ is the dataset with the base dataset $D_b$ and augmented dataset $D_n^{aug}$, and chatGPT_aug is the data augmentation method based on ChatGPT
Parameters: Fine-tuning epochs of base dataset $epoch_b$, fine-tuning epochs of FSL $epoch_f$
  for epoch in $epoch_b$ do
    train($model$, $D_b$)
  end for
  $D_n^{aug}$ = chatGPT_aug($D_n$)
  for epoch in $epoch_f$ do
    train($model$,$D_n^{aug}$)
  end for
```

models and uses transformer decoder blocks (Vaswani et al., 2017) as the model backbone.

During pre-training, ChatGPT is regarded as an unsupervised distribution estimation from a set of samples $X = \{x_1, x_2, ..., x_n\}$, and sample x_i composed of m tokens is defined as $x_i = (s_1, s_2, ..., s_m)$. The objective of pre-training is to maximize the following likelihood:

$$L(x_i) = \sum_{i=1}^{m} \log P(s_i|s_1, ..., s_{i-1}; \theta) \tag{6.1}$$

where θ represents the trainable parameters of ChatGPT. The tokens are represented by token embedding and position embedding:

$$h_0 = x_i W_e + W_p \tag{6.2}$$

where W_e is the token embedding matrix and W_p is the position embedding matrix. Then N transformer blocks are used to extract the features of the sample:

$$h_n = transformer\_blocks(h_{n-1}) \tag{6.3}$$

where $n \in [1, N]$.

Finally, the target token is predicted:

$$s_i = softmax(h_N W_e^T) \tag{6.4}$$

where h_N is the output of top transformer blocks.

After pre-training, the developers of ChatGPT apply Reinforcement Learning from Human Feedback (RLHF)(Ouyang, Wu, Jiang, Almeida, Wainwright, Mishkin, Zhang, Agarwal, Slama, Gray, et al., 2022) to fine-tune the pre-trained language model. The RLHF aligns language models with user intent on a wide range of tasks by fine-tuning them according to human feedback. The RLHF of ChatGPT contains three steps:

Supervised Fine-tuning (SFT): Unlike GPT, GPT-2, and GPT-3, ChatGPT uses labeled data for further training. The AI trainers play as users and AI assistants to build the answers based on prompts. The answers with prompts are used as supervised data for further training of the pre-trained model. After further pre-training, a SFT model can be obtained.

Reward Modeling (RM): Based on the SFT method, a reward model is trained to take in a pair of prompt and response, and output a scalar reward. Human labelers rank the outputs from best to worst to build a ranking dataset. The loss function between two outputs is defined as follows:

$$\operatorname{loss}(\theta_r) = E_{(x,y_w,y_l)\sim D_c}\left[\log\left(\sigma\left(r_{\theta_r}\left(x, y_w\right) - r_{\theta_r}\left(x, y_l\right)\right)\right)\right] \tag{6.5}$$

where θ_r is the parameters of reward model; x is the prompt, y_w is the preferred completion out of the pair of y_w and y_l; D_c is the dataset of human comparisons.

Reinforcement Learning (RL): By using reward models, ChatGPT can be fine-tuned using Proximal Policy Optimization (PPO) (Schulman et al., 2017). In order to fix the performance degradation on public NLP datasets, the RLHF mixes the pretraining gradients into the PPO gradients, which is also known as PPO-ptx:

$$\begin{aligned}&\operatorname{objective}(\phi) = \gamma E_{x\sim D_{\text{pretrain}}}\left[\log\left(\pi_\phi^{\mathrm{RL}}(x)\right)\right] + \\ &E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_{\theta_r}(x, y) - \beta\log\left(\pi_\phi^{\mathrm{RL}}(y \mid x)/\theta_{\mathrm{SFT}}(y \mid x)\right)\right]\end{aligned} \tag{6.6}$$

where π_ϕ^{RL} is the learned RL policy, θ_{SFT} is the supervised trained model, and D_{pretrain} is the pretraining distribution. The γ is the pre-training loss coefficient that controls the strength of pre-training gradients, and the β is the KL (Kullback-Leibler) reward coefficient that controls the strength of the KL penalty.

Compared to previous data augmentation methods, ChatGPT is more suitable for data augmentation for the following reasons:

- ChatGPT is pre-trained on large-scale corpora, so it has a broader semantic expression space, and is helpful to enhance the diversity of data augmentation.

- Since the fine-tuning stage of ChatGPT introduces a large number of manual annotation samples, the language generated by ChatGPT is more in line with human expression habits.

- Through reinforcement learning, ChatGPT can compare the advantages and disadvantages of different expressions and ensure that the generated data are of high quality.

Under the BERT framework, we introduce ChatGPT as the data augmentation tool for few-shot text classification. Specifically, ChatGPT is applied to rephrase each input sentence into six additional sentences, thereby augmenting the few-shot samples.

6.5.3 Few-shot Text Classification

We apply BERT (Devlin et al., 2019b) to train a few-shot text classification model. The output features h of the top layer of BERT can be written as:

$$z = [z_c, z_1, z_2, ..., z_n], \tag{6.7}$$

where z_c is the representation of the class-specific token CLS. For text classification, z_c is usually fed into a task-specific classifier header for final prediction. However, in the FSL scenario, it is difficult to achieve satisfactory performance through BERT fine-tuning because the small scale of few-shot samples will easily lead to over-fitting and lack of generalization ability.

To effectively address the challenge of few-shot text classification, many approaches have been proposed. Generally, there are four categories of methods for few-shot text classification based on large language models: meta-learning, prompt-tuning, model design, and data augmentation. meta-learning refers to the process of *learning to learn* with tasks that update meta-parameters (Lee et al., 2022; Yin, 2020). Prompt-based methods guide large language models to predict correct results by designing templates (Brown et al., 2020; Han et al., 2022; Lester et al., 2021; J. Wang et al., 2022). Model design methods guide the model to learn from few-shot samples by changing the structure of the model (Liao, Liu, Dai, Wu, et al., 2023). Data augmentation uses similar characters (Belinkov & Bisk, 2017), similar word semantics (Alzantot et al., 2018; Mrkšić et al., 2016), or knowledge base (Rezayi, Dai, et al., 2022; Rezayi, Liu, et al.,

2022) to expand samples. Our method directly data augmentation through the language capabilities of large language models, which is a simple and efficient data augmentation method.

Objective Function: Our objective function of few-shot learning consists of two parts: cross entropy and contrastive learning loss. We feed z_c into a fully connected layer, the classifier for the final prediction:

$$\hat{y} = W_c^T z_c + b_c, \tag{6.8}$$

where W_c and b_c are trainable parameters, and take cross-entropy as one of the objective functions:

$$L_{CE} = -\sum_{d \in D'} \sum_{c=1}^{C} y_{dc} \ln \hat{y}_{dc}, \tag{6.9}$$

where C is the output dimension, which is equal to the union of label spaces of the base dataset and novel dataset, and y_d is the ground truth.

Then, to make full use of the prior knowledge in the base dataset to guide the learning of the novel dataset, we introduce the contrastive loss function to make the sample representation of the same category more compact and the sample representation of different categories more separate. The contrastive loss between pairs of samples in the same batch is defined as follows:

$$L_{CL} = -\log \frac{\sum e^{cos(v_i, v_{i'})}}{\sum e^{cos(v_i, v_{i'})} + \sum e^{cos(v_i, v_j)}}, \tag{6.10}$$

where v_i and $v_i^{'}$ are the z_c of samples that belong to the same category; v_i and v_j are the z_c of samples belong to different categories; $cos(\cdot;\cdot)$ is the cosine similarity.

In the BERT fine-tuning stage on the base dataset, we only use cross entropy as the objective function. In the few-shot learning stage, we combine cross entropy and contrastive learning loss as the objective function:

$$L = L_{CE} + \lambda L_{CL}. \tag{6.11}$$

6.5.4 Baseline Methods

In the experiment section, we compare our method with other popular data augmentation methods. For these methods, we use the implementation in

Table 6.2: Data Augmentation and Ablation Study. The BERT + C indicates BERT with contrastive loss.

Data Augmentation	Amazon		Symptoms		PubMed20K	
	BERT	BERT + C	BERT	BERT + C	BERT	BERT + C
Raw	0.734	0.745	0.636	0.606	0.792	0.798
BackTranslationAug	0.757	0.748	0.778	0.747	0.812	0.83
CWAUB(Insert)	0.761	0.750	0.697	0.677	0.802	0.811
CWAUB(Substitute)	0.770	0.757	0.626	0.667	0.815	0.830
CWAUDB(Insert)	0.759	0.762	0.707	0.747	0.796	0.796
CWAUDB(Substitute)	0.787	0.766	0.667	0.646	0.797	0.800
CWAURB(Insert)	0.775	0.768	0.758	0.707	0.815	0.814
CWAURB(Substitute)	0.745	0.730	0.727	0.667	0.782	0.782
CounterFittedEmbeddingAug	0.754	0.741	0.667	0.626	0.805	0.805
InsertCharAugmentation	0.771	0.775	0.404	0.475	0.826	0.831
InsertWordByGoogleNewsEmbeddings	**0.816**	0.794	0.636	0.677	0.786	0.784
KeyboardAugmentation	0.764	0.766	0.545	0.505	0.809	0.815
OCRAugmentation	0.775	0.782	0.768	0.778	0.789	0.789
PPDBSynonymAug	0.691	0.690	0.697	0.758	0.795	0.829
SpellingAugmentation	0.727	0.736	0.697	0.707	0.808	0.811
SubstituteCharAugmentation	0.762	0.768	0.535	0.586	0.816	0.821
SubstituteWordByGoogleNewsEmbeddings	0.729	0.741	0.727	0.727	0.807	0.822
SwapCharAugmentation	0.762	0.766	0.475	0.485	0.797	0.801
SwapWordAug	0.771	0.766	0.687	0.727	0.798	0.794
WordNetSynonymAug	0.805	0.798	0.616	0.758	0.761	0.757
ChatGPT (2-shot)	0.753		0.980		0.748	
AugGPT	**0.816**	**0.826**	**0.889**	**0.899**	**0.835**	**0.835**

open-source libraries including, nlpaug (Ma, 2019) and textattack (Morris et al., 2020).

- **InsertCharAugmentation**. This method inserts random characters at random locations in text, which improves the generalization ability of the model by injecting noise into the data.

- **SubstituteCharAugmentation**. This method randomly replaces selected characters with other ones.

- **SwapCharAugmentation** (Belinkov & Bisk, 2017). This method randomly exchanges two characters.

- **DeleteCharAugmentation**. This method randomly deletes characters.

- **OCRAugmentation**. OCRAugmentation simulates possible errors during OCR recognition. For example, OCR tool may wrongly identify “o" as “o", and wrongly identify “I" as “l".

- **SpellingAugmentation** (Coulombe, 2018). It creates new text by deliberately misspelling some words. The method uses a list of English words that are most likely to be misspelled provided by Oxford Dictionary, for example, misspelling “because" as “becouse".

- **KeyboardAugmentation** (Belinkov & Bisk, 2017). It simulates typo error by replacing randomly selected characters with the adjacent characters in the QWERTY layout keyboard. For example, replacing 'g' with 'r', 't', 'y', 'f', 'h', 'v', 'b' or 'n'.

- **SwapWordAug** (Wei & Zou, 2019a). It randomly exchanges words in text. This method is a submethod of Easy Data Augmentation (EDA) proposed by Wei et al.

- **DeleteWordAug**. DeleteWordAug randomly deletes words in the text, which is also a submethod of EDA.

- **PPDBSynonymAug** (Niu & Bansal, 2018). It replaces words with their synonym in PPDB thesaurus. Synonym replacement can ensure semantic consistency and is suitable for classification tasks.

- **WordNetSynonymAug**. It replaces words with their synonym in WordNet thesaurus.

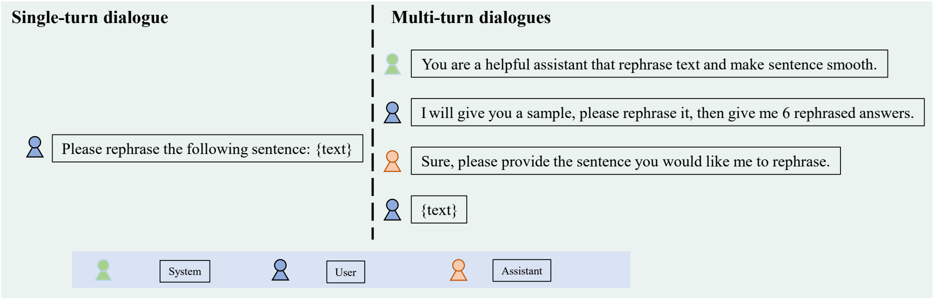
- **SubstituteWordByGoogleNewsEmbeddings** (W. Y. Wang & Yang, 2015). It replaces words with their top-n similar words in the embedding space. The word embeddings used are pre-trained with GoogleNews corpus.

- **InsertWordByGoogleNewsEmbeddings** (Ma, 2019). It randomly selects word from vocabulary of GoogleNews corpus and inserts it the random position of the text.

- **CounterFittedEmbeddingAug** (Alzantot et al., 2018; Mrkšić et al., 2016). It replaces words with their neighbors in counter-fitting embedding space. Compared with GoogleNews word vectors used by SubstituteWordByGoogleNewsEmbeddings, counter-fitting embedding introduces the constraint of synonyms and antonyms, that is, the embedding between synonyms will be pulled closer, and vice versa.

- **ContextualWordAugUsingBert(Insert)** (Kobayashi, 2018; Kumar et al., 2020). This method uses BERT to insert words based on context, that is, add $< mask >$ token at random position of the input text, and then let BERT predict the token at that position.

- **ContextualWordAugUsingDistilBERT(Insert)**. This method uses DistilBERT to replace BERT for prediction, and the rest is the same as ContextualWordAugUsingBert(Insert).



Please rephrase the following sentence: {text}

You are a helpful assistant that rephrase text and make sentence smooth.

I will give you a sample, please rephrase it, then give me 6 rephrased answers.

Sure, please provide the sentence you would like me to rephrase.

{text}

Figure 6.2: Single-turn dialogue and multi-turn dialogues prompt

- **ContextualWordAugUsingRoBERTA(Insert)**. This method uses RoBERTA to replace BERT for prediction, and the rest is the same as ContextualWordAugUsingBert(Insert).

- **ContextualWordAugUsingBert(Substitute)**. This method (Kobayashi, 2018; Kumar et al., 2020) uses BERT to replace words based on context, that is, replace randomly selected words in text with $< mask >$ token, and then let BERT predict the token at that position.

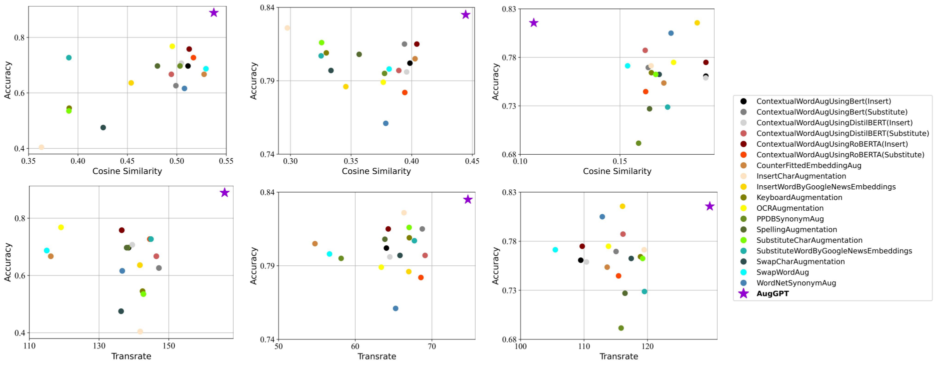
- **ContextualWordAugUsingDistilBERT(Substitute)**. This method uses DistilBERT to replace BERT for prediction, and the rest is the same as ContextualWordAugUsingBert(Substitute).

- **ContextualWordAugUsingRoBERTA(Substitute)**. This method uses RoBERTA to replace BERT for prediction, and the rest is the same as ContextualWordAugUsingBert(Substitute).

- **BackTranslationAug**. The method (Sennrich et al., 2016) translates the text into German and then into English, resulting in a new text that is different from the original but has the same semantics. We use wmt19-en-de and facebook/wmt19-de-en language translation models (Ng et al., 2020) developed by Facebook for translation.

6.5.5 Prompt Design

We have designed prompts for single-turn dialogue and multi-turn dialogues. The prompts are shown in Fig 6.2. The Amazon dataset use the multi-turn dialogues prompt for data augmentation. The Symptoms and PubMed20K use the single-turn dialogue prompt for data augmentation.



(a) Symptoms (b) PubMed20K (c) Amazon

Figure 6.3: We employed two evaluation metrics to assess the faithfulness and compactness of our newly augmented data. The top left plot displays the cosine similarity metric and final accuracy of all data augmentation methods on the Symptoms dataset, and the bottom left plot shows the TransRate metric and final accuracy of all data augmentation methods on the Symptoms dataset. In the middle and bottom panels, we plotted the cosine similarity and TransRate values of all data augmentation methods on the Amazon and PubMed20K datasets, respectively. On the right side of the picture, we listed all the augmented methods with different colors and shapes.

6.5.6 Evaluation Metrics

We employed cosine similarity and TransRate(L.-K. Huang et al., 2022) as metrics to assess the faithfulness (i.e., whether the generated data samples are close to the original samples) and compactness (i.e., whether samples of each class are compact enough for good discrimination) of the augmented data.

6.5.7 Embedding Similarity

To evaluate the semantic similarity between the samples generated by data augmentation methods and actual samples, we adopt embedding similarity between the generated samples and the actual samples of the test dataset. Some of the most common similarity metrics include Euclidean distance, cosine similarity and dot product similarity. In this study, we select cosine similarity to capture the distance relationship in the latent space. The cosine similarity measures the cosine value of the angle between two vectors. This value increases when two vectors are more similar, and is bounded by a range between 0 and 1. Since the pre-trained language models without fine-tunning poorly to capture semantic meaning, we fine-tunning the pre-trained BERT on base dataset by

BERT-flow (B. Li et al., 2020) method, and finally apply the fine-tunned BERT to get smaple embedding. The cosine similarity metric is commonly used in NLP (J. Wang & Dong, 2020) and we follow this convention.

$$\cos(\theta) = \frac{A \cdot B}{\|A\|_2 \|B\|_2}, \tag{6.12}$$

where A and B denote the two embedding vectors in comparison, respectively.

6.5.8 TransRate

TransRate is a metric that quantifies transferability based on the mutual information between the features extracted by a pre-trained model and their labels, with a single pass through the target data. The metric achieves a minimum value when the data covariance matrices of all classes are identical, making it impossible to distinguish between the data from different classes and preventing any classifier from achieving better than random guessing. Thus, a higher TransRate could indicate better learnability of the data. More specifically, knowledge transfer from a source task T_s to a target task T_t is measured as shown below:

$$TrR_{T_s \to T_t}(g) = H(Z) - H(Z|Y), \tag{6.13}$$

where Y represents the labels of augmented examples, and Z denotes the latency embedding features extracted by the pre-trained feature extractor g. TrR means the TransRate value. $H(\cdot)$ denotes the Shannon entropy(Cover, 1999).

6.5.9 Direct Classification Performance by ChatGPT

An interesting and important question about the utilization of ChatGPT for text data augmentation would be how ChatGPT will perform when directly applied to FSL downstream tasks. Thus, we developed tailored prompts for ChatGPT to perform the classification tasks with integrated the API for prompting. For the Symptoms dataset, we employed the following prompt instruction: "Given a person's health description or symptom, predict the corresponding illness from the following categories: CLASSES." Additionally, we used "Description: DESCRIPTION. Typically, this symptom corresponds to CLASS" as the prompt for each example in the dataset. In this way, We can include few-shot examples (in this work, we used two) to facilitate the model's adaptation to downstream tasks. We used similarly-designed prompt instructions for the other two tasks and the corresponding example prompt to implement the few-shot in-context learning by ChatGPT.

6.6 Experiment Results

In our experiments, we use BERT as the base model. Firstly, we train our model on the base dataset to produce the pre-trained model. Then we fine-tune the model with the combination of few-shot samples and the augmented samples generated from various data augmentation methods. Specifically, in all three FSL tasks, we perform 2-shot learning, i.e., there would be two real samples used for each class in the target domain. Afterward, We use those samples to fine-tune the pre-trained models. To evaluate the effectiveness of different data augmentation methods, we apply two different settings. The first one is the vanilla BERT model. In the second setting, we add a contrastive loss to the training objective function. In our experiments on the Symptoms dataset, we use a batch size of 8 for 150 epochs, set the maximum sequence length to 25, λ as 1, and use a learning rate of 4e-5. In our experiments on the PubMed20K dataset, we adopt the same training configuration, with the maximum sequence length set to 40. For all three tasks, we will generate six augmented samples per class. Examples of the augmented samples generated by AugGPT and other selected baseline methods can be found in the appendix. Codes and the three benchmark datasets can be found at https://github.com/yhydhx/AugGPT.

6.6.1 Classification Performance Comparison

Table 6.2 shows the accuracy of different data augmentation methods. As shown in Table 6.2, AugGPT achieves the highest accuracy for Amazon, Symptoms and PubMed20K datasets. For the Amazon dataset, AugGPT and InsertWordByGoogleNewsEmbeddings achieve the best performance for BERT, and AugGPT achieve the best performance for BERT with contrastive loss. In the PubMed20K dataset, AugGPT achieves 83.5% accuracy for both BERT and BERT with contrastive loss, whereas without data augmentation, the accuracy values are only 79.2% and 79.8%, respectively. For the Symptoms dataset, the accuracy for BERT downstream augmentation is only 63.6%, and 60.6% with contrastive loss. However, our AugGPT approach significantly improves the accuracy to 88.9% and 89.9%, respectively. These results suggest that data augmentation using ChatGPT is more effective in enhancing the performance of machine learning models in various applications.

6.6.2 Evaluation of Augmented Datasets

In addition to the classification accuracy, we evaluate the augmented data in the latent space and visualize the results in Fig 6.3. Latent embeddings are evaluated using cosine similarity and the TransRate metric (see section 6.5.6 for more details). The horizontal axis represents the cosine similarity values and Transrate values, and the vertical axis describes the classification accuracy. Since embedded similarity measures the similarity between the generated data and the test dataset, high similarity means that the generated data are close to real input data and with higher faithfulness and compactness. Higher TransRate indicates better learnability of the data. Therefore, a higher TransRate score indicates that the augmented data are of higher quality. The most ideal candidate method should be positioned at the top-right of the visualization. As shown in Fig 6.3, AugGPT produces high-quality samples in terms of both faithfulness and compactness on the Symptoms dataset and the PubMed20K dataset. On the open-domain Amazon dataset, AugGPT also produces high-quality samples with a higher TransRate.

6.6.3 Performance Comparison with ChatGPT

Furthermore, we used ChatGPT to directly perform the downstream text data classification tasks under a 5-shot learning scheme. We used in-house designed instructions with few-shot in-context examples to prompt ChatGPT as described in 4.7. The performance of ChatGPT for the downstream tasks is listed in Table 2. The result reveals that state-of-the-art large language models such as ChatGPT tend to perform better on relatively easier tasks, for example, identifying symptoms according to a one-sentence description. However, when it comes to complicated tasks such like PubMed, model fine-tuning is still needed and could achieve better performance compared to few-shot prompts.

6.7 Conclusion and Discussion

In this paper, we proposed a novel data augmentation approach for few-shot classification. Unlike other methods, our model expands the limited data at the semantic level to enhance data consistency and robustness, which results in a better performance than most of the current text data augmentation methods. With the advancement of LLM and its nature of a multi-task learner (Radford et al., 2019), we envision that a series of tasks in NLP can be enhanced or even replaced in a similar fashion.

Although AugGPT has shown promising results in data augmentation, it has certain limitations. For example, when recognizing and augmenting medical texts, AugGPT may produce incorrect augmentation results due to the lack of domain knowledge of ChatGPT. In future works, we will investigate adapting the general-domain LLMs, such as ChatGPT, to domain-specific data, such as medical texts, via model fine-tuning, in-context learning (prompt engineering), knowledge distillation, style transfer, etc.

AugGPT has demonstrated that the augmentation results can effectively improve the performance of the downstream classification task. A promising direction for future research is to investigate AugGPT against a wider range of downstream tasks. For example, given the strong ability of ChatGPT to extract key points and understand sentences, it can be utilized in tasks such as text summarization. Specifically, ChatGPT might be valuable for domain-specific science paper summarization (Cai et al., 2021) and clinical report summarization (Cai et al., 2022). Publicly available domain-specific science paper summarization datasets and clinical report datasets are rare and often provided at small scales due to privacy concerns and the need for expert knowledge to generate annotated summaries. However, ChatGPT could address this challenge by generating diverse augmented summarization samples in different representation styles. The data generated from ChatGPT are typically concise, which can be valuable for further enhancing the generalization capabilities of the trained model.

The dramatic rise of generative image models such as DALLE2 (Ramesh et al., 2022) and Stable Diffusion (R. Rombach et al., 2022) provides opportunities for applying AugGPT to few-shot learning tasks in computer vision. For example, accurate language descriptions may be used to guide the generative model to generate images from text or to generate new images based on existing images as a data augmentation method for few-shot learning tasks, especially when combined with efficient fine-tuning methods (E. J. Hu et al., 2021; Ruiz et al., 2022) such as LoRA for Stable Diffusion. Thus, prior knowledge from a large language model can facilitate faster domain adaptation and better few-shot learning of generative models in computer vision.

Recent research shows that large language models (LLMs), such as GPT-3 and ChatGPT, are capable of solving Theory of Mind (ToM) tasks, which were previously thought to be unique to humans (Kosinski, 2023). While the ToM-like capabilities of LLMs may be an unintended byproduct of improved performance, the underlying connection between cognitive science and the human brain is an area ripe for exploration. Advancements in cognitive and brain science can also be used to inspire and optimize the design of LLMs. For example, it has been suggested that the activation patterns of the neurons in the BERT model and those in the human brain networks may share similarities and could be coupled together(X. Liu et al., 2023). This presents a promising new direction for developing LLMs by utilizing prior knowledge from brain science. As researchers continue to investigate the connections between LLMs and the human brain, we may discover new means to enhance the performance and capabilities of AI systems, leading to exciting breakthroughs in the field.

Chapter 7

An Autonomous GPT for Alzheimer's Disease Infodemiology

7.1 Overview

Inspired by AutoGPT, the state-of-the-art open-source application based on the GPT-4 large language model, we develop a novel tool called AD-AutoGPT which can conduct data collection, processing, and analysis about complex health narratives of Alzheimer's Disease in an autonomous manner via users' textual prompts. We collated comprehensive data from a variety of news sources, including the Alzheimer's Association, BBC, Mayo Clinic, and the National Institute on Aging since June 2022, leading to the autonomous execution of robust trend analyses, intertopic distance maps visualization, and identification of salient terms pertinent to Alzheimer's Disease. This approach has yielded not only a quantifiable metric of relevant discourse but also valuable insights into public focus on Alzheimer's Disease. This application of AD-AutoGPT in public health signifies the transformative potential of AI in facilitating a data-rich understanding of complex health narratives like Alzheimer's Disease in an autonomous manner, setting the groundwork for future AI-driven investigations in global health landscapes.

7.2 Background

Alzheimer's Disease (AD), a progressive neurodegenerative disorder, remains one of the most pressing public health concerns globally in the 21st century (Avramopoulos, 2009; Dartigues, 2009). This disease, characterized by cognitive impairments such as memory loss, predominantly affects aging populations, exerting an escalating burden on global healthcare systems as societies continue to age (Post, 2000). The significance of AD is further magnified by the increasing life expectancy globally, with the disease now recognized as a leading cause of disability and dependency among older people (Hinton & Levkoff, 1999). Consequently, AD has substantial social, economic, and health system implications, making its understanding and awareness of paramount importance (Rice et al., 1993; Y. Zhao et al., 2008).

Despite the ubiquity and severity of AD, a gap persists in comprehensive, data-driven public understanding of this complex health narrative. Traditionally, public health professionals have to rely on labor-intensive methods such as web scraping, API data collection, data postprocessing, and analysis/synthesis to gather insights from news media, health reports, and other textual sources (Bacsu, Fraser, et al., 2022; Mavragani, 2020; Y. Zhang, Lyu, et al., 2021). However, these methods often necessitate complex pipelines for data gathering, processing, and analysis. Moreover, the sheer scale of global data presents an ever-increasing challenge, one that demands a novel, innovative approach to streamline these processes and extract valuable, actionable insights efficiently and automatically. In addition, the technical expertise required for developing data processing and analysis pipelines significantly limits the access and engagement of the broader public health community.

AutoGPT (Richards, 2023) is an experimental open-source application that harnesses the capabilities of large language models (LLMs) such as GPT-4 (OpenAI, 2023) and ChatGPT (Y. Liu et al., 2023) to automate and optimize the analytical process. With its advanced linguistic understanding and autonomous operation, AutoGPT simplifies complex data pipelines, facilitating comprehensive analyses of vast datasets with simple textual prompts. This tool transcends traditional limitations, unlocking the potential of LLMs for autonomous data collection, processing, summarization, analysis, and synthesis.

In this study, we modify the AutoGPT architecture into public health applications and develop AD-AutoGPT to analyze a multitude of news sources, including the Alzheimer's Association, BBC, Mayo Clinic, and the National Institute on Aging, focusing on discourse since June 2022. We are among the pioneers in integrating AutoGPT into public health informatics, adapting this transformative AI tool into the public health domain to elucidate the complex narrative surrounding Alzheimer's Disease. This research underlines the enormous potential of autonomous LLMs in global health research, paving the way for future AI-assisted investigations into various health-related domains.

We summarize our key contributions below:

- Inspired by AutoGPT, we develop a novel LLM-based tool called AD-AutoGPT, which can generate data collection, processing, and analysis pipeline in an autonomous manner based on users' textual prompts. More specifically, we adapt AD-AutoGPT to the public health domain to showcase its great potential of autonomous pipeline generation to understand the complex narrative surrounding Alzheimer's Disease.

- While AutoGPT is an effective autonomous LLM-based tool, it has lots of limitations when applying it on AD Infodemiology during the process of public health information retrieval, text-based information extraction, text summarization, summary analysis, and visualization.

  To overcome AutoGPT's limitations for the AD Infodemiology task, AD-AutoGPT provides the following improvements: 1) specific prompting mechanisms to improve the efficiency and accuracy of AD information retrieval; 2) a tailored spatiotemporal information extraction functionality; 3) an improved text summarization ability; 4) an in-depth analysis ability on generated text summaries; and 5) an effective and dynamic visualization capability.

- We show that AD-AutoGPT transforms the traditional labor-intensive data collection, processing, and analysis paradigm into a prompt-based automated, and optimized analytical framework. This has allowed for efficient, comprehensive analysis of numerous news sources related to Alzheimer's Disease.

- Through AD-AutoGPT, we have provided a case study for detailed trend analysis, intertopic distance mapping, and identified salient terms related to Alzheimer's Disease from four AD-related new sources. This contributes significantly to the existing body of knowledge and facilitates a nuanced understanding of the disease's discourse in public health.

- Our research underlines the capacity of AD-AutoGPT to facilitate data-driven public understanding of complex health narratives, such as Alzheimer's Disease, which is of paramount importance in an aging global society.

- The methodologies and insights from our work provide a foundation for future AI-assisted public health research. Our AD-AutoGPT pipeline is extendable to other topics in public health or even other domains. This work paves the way for comprehensive and efficient investigations into various domains.

7.3 Related Works

7.3.1 Large Language Models

Large language models (LLMs), with their origins in Transformer-based pretrained language models (PLMs) such as BERT (Devlin et al., 2018) and GPT (Radford et al., 2018), have substantially transformed the field of natural language processing (NLP). LLMs have superseded previous methods such as Recurrent Neural Network (RNN) based models, leading to their widespread adoption across various NLP tasks (Y. Liu et al., 2023; L. Zhao, Zhang, et al., 2023). Furthermore, the emergence of very large language models such as GPT-3 (Brown et al., 2020), Bloom (Scao et al., 2022), GPT-4 (OpenAI, 2023), PaLM (Chowdhery et al., 2022), and PaLM-2 (Anil et al., 2023) demonstrates a clear trend towards even more sophisticated language understanding capabilities.

These models are designed to learn accurate contextual latent feature representations from input text (Kalyan et al., 2021), which can then be employed in a variety of applications, including question answering, information extraction, sentiment analysis, text classification, and text generation. The innovative technique of reinforcement learning from human feedback (RLHF) (Ziegler et al., 2019) has been used to further align LLMs with human preferences, which has found applications in Artificial General Intelligence (AGI) models such as InstructGPT (Ouyang, Wu, Jiang, Almeida, Wainwright, Mishkin, Zhang, Agarwal, Slama, Ray, et al., 2022), Sparrow (Glaese et al., 2022), and ChatGPT (Y. Liu et al., 2023). More recently, GPT-4 has significantly advanced the state-of-the-art of language models, opening up new opportunities for LLM applications.

Other than the applications in NLP domain, LLMs also show promising results and significant impacts in other disciplines such as biology (Agathokleous et al., 2023), geography (Mai, Cundy, et al., 2022; Mai et al., 2023), agriculture (Lu et al., 2023), education (Kasneci et al., 2023; Latif et al., 2023), medical and health care (Dave et al., 2023; Z. Liu et al., 2023), and so on.

7.3.2 Public Health Infodemiology

Infodemiology (Eysenbach, 2002) is a field that studies the determinants and distribution of information on the internet or in a population, with the goal of informing public health and public policy (Eysenbach, 2002; Mavragani, 2020). The term combines "information" and "epidemiology" and is a recognized approach in public health informatics, providing insights into health-related behaviors and perceptions. It plays a crucial role in monitoring and managing the information epidemic ("infodemic") associated with major public health crises.

For example, Piamonte et al. (Piamonte et al., 2022) analyzed global search queries for Alzheimer's disease (AD) using Google Trends data, comparing this online interest (Search Volume Index) with measures of disease burden. The study revealed that search behavior and interest in AD were influenced by factors like news about celebrities with AD and awareness months, and also highlighted potential correlations between this online interest and socioeconomic development.

With the rise of the internet and digital technologies, infodemiology provides a vital lens to examine the flow of health information and misinformation, helping public health practitioners develop effective communication strategies and interventions (Mackey et al., 2022; Zielinski, 2021). In the context of Alzheimer's disease, understanding online behaviors and interests via infodemiology can help enhance public awareness, correct misconceptions, and inform preventative and management strategies for the disease (Bacsu, Cammer, et al., 2022; Piamonte et al., 2022).

7.3.3 AutoGPT and LLM Automation

The development and use of AutoGPT, LangChain[2], and many other automation techniques for LLMs represent a significant advancement in the field of NLP and artificial intelligence. AutoGPT builds on the successes of large language models like GPT-3 and GPT-4, but takes automation a step further by providing a more user-friendly interface for non-expert users (Richards, 2023).

With AutoGPT, complex tasks such as data collection, data cleaning, analysis, and even the generation of human-like text can be completed using straightforward prompts, removing the need for extensive coding or data science expertise. This has the potential to democratize access to powerful language model technology, opening up new possibilities for research and application in a wide range of fields, including public health.

Recent studies (Fezari & Ali-Al-Dahoud, n.d.; G. T. Zhao, n.d.) have highlighted the potential of AutoGPT and similar tools for automating the retrieval and analysis of large datasets. For example, with a well-formulated query, AutoGPT can be directed to crawl through a wide array of online platforms, collecting and analyzing comments, discussions, and posts pertaining to vaccines. The system would subsequently generate a summarizing report, outlining major themes of public opinion and prevalent misconceptions, thereby providing valuable insights for public health officials in formulating targeted communication and intervention strategies.

[2] https://python.langchain.com/en/latest/index.html

In the context of infodemiology, AutoGPT can automate the process of analyzing online health information trends, which traditionally involves extensive manual effort. Specifically, it can efficiently scan and interpret internet data, track the spread of health information and misinformation, assess public reaction to health policies or events, and potentially predict future trends.

7.3.4 Improving Autonomous LLM-based Tools for Public Health

While recognizing the potential of autonomous large language models (LLMs) like AutoGPT in public health research and practice, we identified certain limitations in their current state that may hinder their efficacy in particular use cases, such as infodemiology. By tailoring these tools to the specific needs of public health professionals, we aim to enhance their utility in these contexts.

Firstly, despite AutoGPT's extensive searching capabilities, its ability to acquire specialized information quickly and precisely, for instance, about Alzheimer's disease (AD), can be somewhat limited. In response to this, we have integrated specific prompting mechanisms in our model, AD-AutoGPT. These tailored prompts direct AD-AutoGPT to gather data from a select list of authoritative websites relevant to AD, which enhances the efficiency and relevance of information acquisition.

Secondly, Our AD-AutoGPT model also addresses the challenge AutoGPT faces in extracting critical details such as the time and place of news events from articles accurately. AD-AutoGPT uses web-crawling scripts to extract accurate timestamps from news pieces, and employs geo-location libraries such as geopy ("GitHub - geopy/geopy: Geocoding library for Python. — github.com", n.d.) and geopandas ("GitHub - geopandas/geopandas: Python tools for geographic data — github.com", n.d.) to retrieve precise location information from texts.

Thirdly, depth of analysis is another area where AutoGPT could benefit from further refinement. Owing to the token limit in models like ChatGPT, AutoGPT's analysis is often restricted to the first 4096 tokens (Y. Liu et al., 2023). Consequently, it might miss core content or important details. To overcome this limitation, AD-AutoGPT segments the text, vectorizes it, and then processes these chunks independently. It creates summaries for each of these segments and then amalgamates these summaries to create a comprehensive representation of the news article.

Fourthly, AutoGPT's current capabilities, while useful, lack the capacity to conduct an in-depth analysis of the generated summaries. The synthesized data can still be redundant and may not accurately capture the most essential information. In contrast, AD-AutoGPT applies Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to extract the most pertinent keywords from the text summaries, offering users a succinct understanding of the central themes in the Alzheimer's disease domain.

Lastly, while AutoGPT is effective at generating text-based information, it lacks robust visualization capabilities. Addressing this limitation, AD-AutoGPT integrates dynamic visualization techniques, creating plots of news occurrences over time, highlighting locations where news events are happening, and even illustrating the evolution of research keywords over time.

AD-AutoGPT is refined through the application of domain-specific knowledge and technical adjustments to optimize its relevance and effectiveness for public health researchers and practitioners. As a result, AD-AutoGPT is faster and more efficient in its operations compared to the original AutoGPT, highlighting the advantages of tailoring autonomous LLM-based tools for specific use cases in public health.

7.4 Method

In this section, we will introduce AD-AutoGPT, an LLM-based tool we developed to automate the process of Alzheimer's Disease Infodemiology. AD-AutoGPT uses the Langchain framework to realize the connection with GPT-4 and ChatGPT API, and establish an LLM-based autonomous framework with a chain of thinking mode for Alzheimer's disease. This is a model that can automatically search for the latest news, extract meaningful spatio-temporal data, summarize the news, analysis news content, and visualize analysis results. The overall framework of AD-AutoGPT is shown in Figure 7.1. We construct an instruction library that contains a set of possible commands/tools we have developed to achieve the public health infodemiology task. A prompt shown in Figure 7.2a is designed to facilitate LLMs to identify usable tools from the instruction library and form a data processing pipeline that demonstrates the process of thinking. AD-AutoGPT's ability of "translating" natural language prompts to real data processing pipeline is similar to the idea of semantic parsing used in traditional question answering literature (Berant et al., 2013; Liang et al., 2017; Mai et al., 2021), which aims at translating a natural language question into an executable query for a given database or knowledge base. However, the difference is that semantic parsing is only able to generate rather simple executable queries on a well-defined knowledge base while our AD-AutoGPT can handle much more complex real-world tasks such as searching and collecting news from Google, analyzing new contents, and visualizing topic trends

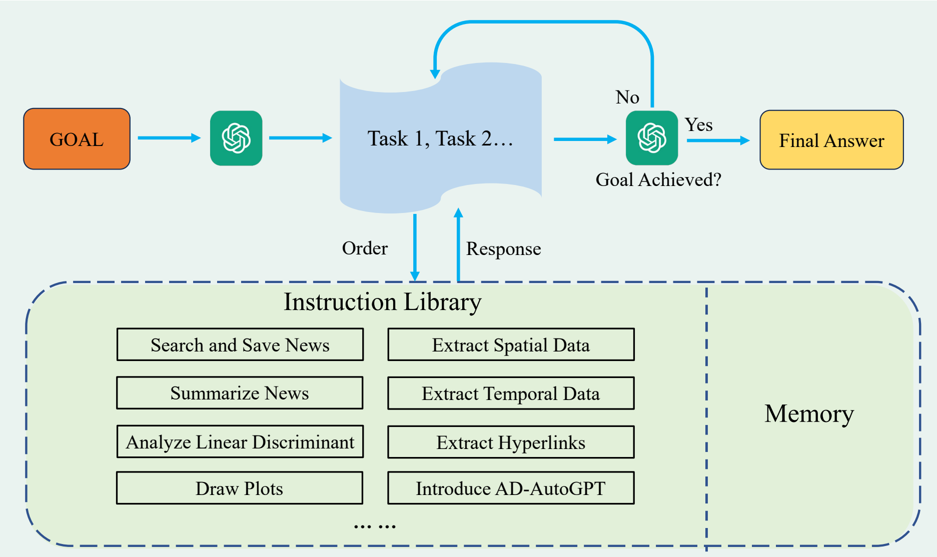


Figure 7.1: The basic framework of AD-AutoGPT. The instruction library contains a set of possible commands we have developed to complete the public health infodemiology task. These commands can also be expanded in the future. In order to achieve the goal, AD-AutoGPT will access GPT-4 and divide the final goal into several smaller tasks, and then solve small tasks step-by-step by choosing the most appropriate command for the sub-task in the instruction library. After thinking and judging, if the final goal has not been achieved, AD-AutoGPT will continue to split the task and execute the command. If the final goal has been achieved, AD-AutoGPT will return the final answer.

and spatial-temporal distributions of news. Below we will introduce the workflow of AD-AutoGPT and the basic principles of the algorithms used in the workflow in detail.

7.4.1 Overall Framework

Our primary goal is to learn from the chain thinking mode of AutoGPT to realize the automatic collection and summary of Alzheimer's disease news. To achieve this goal, the power of LLMs must be used. Advanced LLMs such as ChatGPT and GPT-4 have brought earth-shaking changes to the NLP domain, and we see the potential advantages of LLMs for the public health field.

The overall framework is shown in Figure 7.1. For the target task, AD-AutoGPT will use ChatGPT or GPT-4 to divide the target task into several small tasks and process them separately. We provide AD-AutoGPT with an instruction library which contains customized functions/tools including:

1. **Search and Save News**, which utilizes Google API to search for the latest news posted on authoritative websites and save the URLs on a local device;

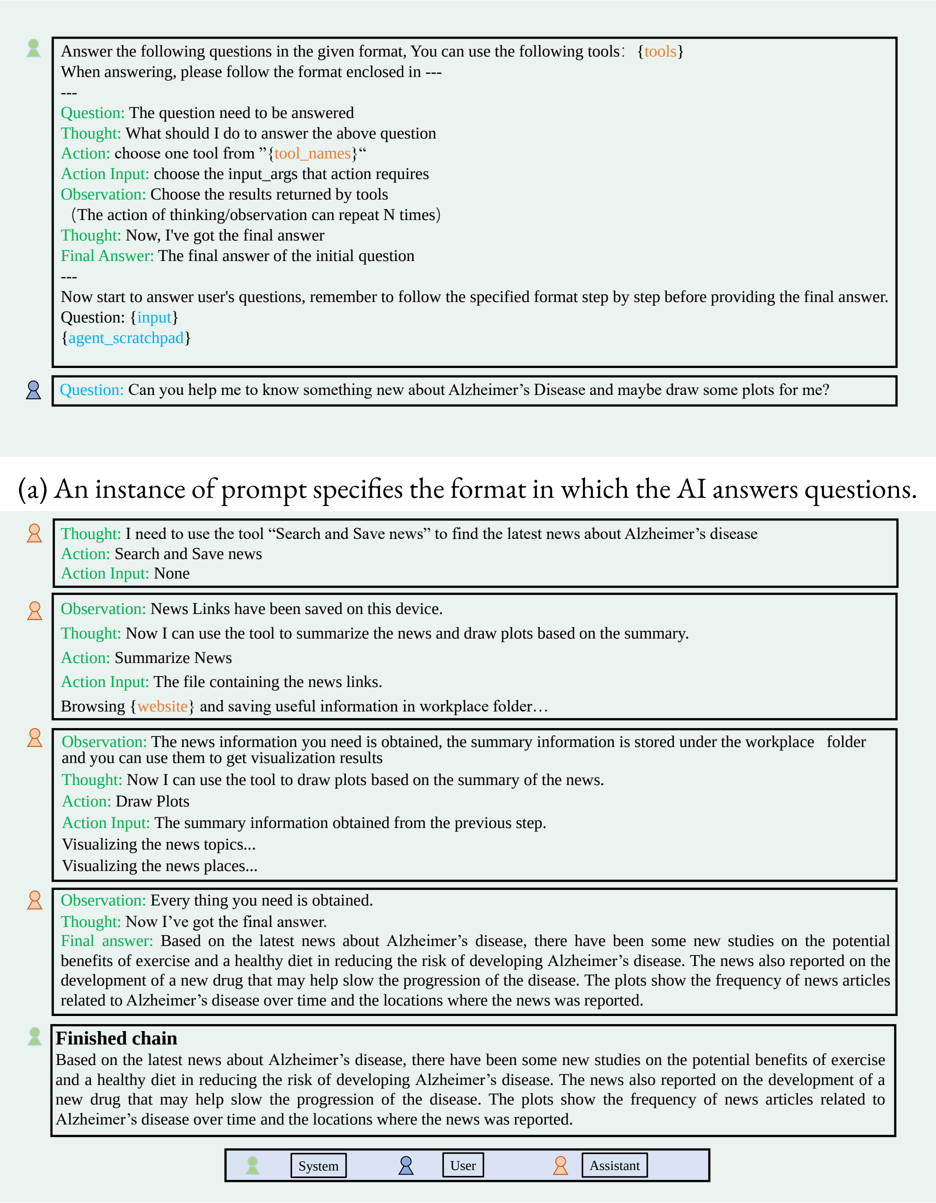
2. **Summarize News**, which uses ChatGPT or GPT-4 to summarize the main content of one piece of news and extract the spatial-temporal information of each stored news;

3. **Visualize Results**, which will draw all the results for visualization, and will also display the results of the LDA analysis of the news content.

After operating every small task choosing from these tools, AD-AutoGPT will judge whether the overall goal has been achieved according to the running results of the function, or it needs to think again and solve the next small problem. Chain thinking is realized through such a pattern. If during the process AD-AutoGPT thinks that the system has reached the initial goal, the system will exit and return a final answer to the initial question.

7.4.2 Designing Prompts to Implement Chain of Thoughts

A prompt example can be seen in Figure 7.2a and the model thinking process of AD-AutoGPT is shown in Figure 7.2b. According to the input, this prompt has four parts in the task process which are question, thought, action, and action input.



```
Answer the following questions in the given format, You can use the following tools: {tools}
When answering, please follow the format enclosed in ---
---
Question: The question need to be answered
Thought: What should I do to answer the above question
Action: choose one tool from "{tool_names}"
Action Input: choose the input_args that action requires
Observation: Choose the results returned by tools
 (The action of thinking/observation can repeat N times)
Thought: Now, I've got the final answer
Final Answer: The final answer of the initial question
---
Now start to answer user's questions, remember to follow the specified format step by step before providing the final answer.
Question: {input}
{agent_scratchpad}
```

```
Question: Can you help me to know something new about Alzheimer's Disease and maybe draw some plots for me?
```

(a) An instance of prompt specifies the format in which the AI answers questions.

```
Thought: I need to use the tool "Search and Save news" to find the latest news about Alzheimer's disease
Action: Search and Save news
Action Input: None
```

```
Observation: News Links have been saved on this device.
Thought: Now I can use the tool to summarize the news and draw plots based on the summary.
Action: Summarize News
Action Input: The file containing the news links.
Browsing {website} and saving useful information in workplace folder…
```

```
Observation: The news information you need is obtained, the summary information is stored under the workplace   folder
and you can use them to get visualization results
Thought: Now I can use the tool to draw plots based on the summary of the news.
Action: Draw Plots
Action Input: The summary information obtained from the previous step.
Visualizing the news topics...
Visualizing the news places...
```

```
Observation: Every thing you need is obtained.
Thought: Now I've got the final answer.
Final answer: Based on the latest news about Alzheimer's disease, there have been some new studies on the potential
benefits of exercise and a healthy diet in reducing the risk of developing Alzheimer's disease. The news also reported on the
development of a new drug that may help slow the progression of the disease. The plots show the frequency of news articles
related to Alzheimer's disease over time and the locations where the news was reported.
```

Finished chain

Based on the latest news about Alzheimer's disease, there have been some new studies on the potential benefits of exercise and a healthy diet in reducing the risk of developing Alzheimer's disease. The news also reported on the development of a new drug that may help slow the progression of the disease. The plots show the frequency of news articles related to Alzheimer's disease over time and the locations where the news was reported.

(b) An example of AI thinking and calling functions to solve user problems

Figure 7.2: The prompt of AD-AutoGPT, the AI assistant will answer the question based on the given format and can use the specified functions. In the prompt, tools represent the functions that AD-AutoGPT can call, including tool_names, tool_descriptions and so on.

1. **Question** is the problem that AI needs to solve.

2. **Thought** is the idea and thought process of AI for this problem.

3. **Action** is the operation selected by AI after thinking which AI thinks is most suitable for solving the current task.

4. **Action input** is used as the input of the function.

For output, a prompt has three parts which are observation, thought, and final answer.

1. **Observation** is the output of the function to inspire AI's next thinking.

2. **Thought** shows the results of AI's thinking about Observation.

3. The **final answer** is the judgment of the result. If the AI thinks that the current result can fully answer the initial question, the AI will return the final answer. Otherwise, it will continue to think and call other functions.

The last part of the prompt is the question entered by the user, such as the question in Figure 7.2a, "*Can you help me to know something new about Alzheimer's Disease and maybe draw some plots for me?*". AI will decompose the complex target tasks proposed by users into several simple tasks, thus inspiring a chain of thoughts. And the thinking process of AI can be seen in Figure 7.2b

Owing to this set of prompts, we can ensure that the thinking logic of AD-AutoGPT does not deviate from the right track and make the whole chain of thoughts visible to users.

7.4.3 Text Summary

To achieve the purpose of extracting the most critical information from a large amount of news text, AD-AutoGPT performs new text summary and LDA topic modeling.

The text summary is mainly achieved by accessing ChatGPT or GPT-4 API. Owing to the powerful text summarization ability of GPT-4, AD-AutoGPT can make more efficient use of text than other models. AD-AutoGPT traverses the saved news URLs one by one, and then saves the text from the website by calling the web crawler scripts. Next, it uses ChatGPT or GPT-4 to summarize the news text. It is worth mentioning that because LLMs have a token limit, all the text here will be pre-processed first, and then be summarized. More specifically, since GPT-4 has a limit on the number of tokens, in order to summarize the complete news text, we use the map_reduce method to process it (Richards, 2023).

7.4.4 Spatiotemporal Information Extraction

Next, AD-AutoGPT will perform spatiotemporal information extraction on the collected news articles. The temporal information can be easily extracted from news metadata while extracting place mentions from news articles is a kind of oral. Here, we adopt the geoparsing approach (Gritta et al., 2018; Karimzadeh et al., 2019) which first recognizes place names from raw text, so-called toponym recognition (Mai, Cundy, et al., 2022; J. Wang et al., 2020) and then link the recognized place names to a specific geographic entity in an existing gazetteer or geospatial knowledge graphs (Ahlers, 2013; Mai, Hu, et al., 2022), so-called toponym resolution (Ju et al., 2016), so that the spatial footprints (i.e., geographic coordinates) of these places can be obtained. More specifically, we use GeoText[3], a python-based geoparsing tool to achieve this goal.

7.4.5 LDA Analysis

Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is a probabilistic topic model. LDA can give a probability distribution of topics of each document in the corpus. By analyzing a batch of document sets and extracting their topic distributions, topic clustering can be performed according to the topic distribution. LDA is a typical bag-of-words model, that is, a document is interpreted as a set of words, and there is no sequential relationship among words. In addition, a document can contain multiple topics, and each word in the document is assumed to be generated by one of the topics. LDA is an unsupervised learning method that does not require a manually labeled training set during training but only needs a document set and the total number of topics K. In addition, another advantage of LDA is that every topic is associated with a set of most frequent keywords which can be used to interpret this topic.

In short, AD-AutoGPT uses LDA topic modeling to discover the topics for the summary text of each piece of collected news, For each topic, the keyword with the highest frequency of occurrence and the highest weight will be displayed to the user.

7.5 Case Study and Experimental Results

7.5.1 Alzheimer's Disease News Information Retrieval

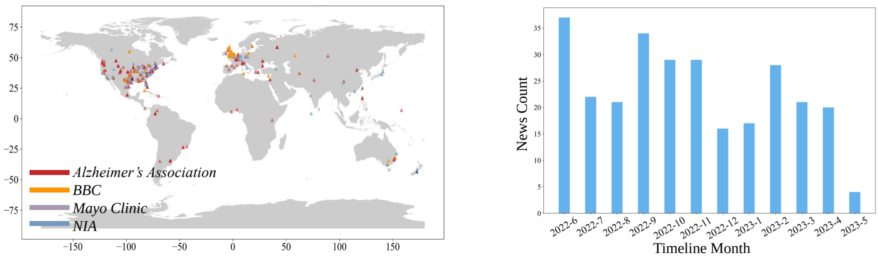
The effectiveness of our proposed AD-AutoGPT is mainly verified on the data provided by the most authoritative websites reporting Alzheimer's disease, which are Alzheimer's Association, BBC, National Institute of Aging, and Mayo

[3] https://github.com/elyase/geotext

Clinic. By using the prompt shown in Figure 7.2a, we are able to instruct the LLM (e.g., ChatGPT or GPT-4) to search for the right tool in our instruction library – *Search and Save News* to achieve the first news data collection step.

We have collected 277 news in total from these four websites in the period of last year. On this actual news dataset, we validate the functions of AD-AutoGPT for text extraction, text summarization, spatio-temporal-data analysis, hot topics analysis, and result visualization. In this process, the time and location of the news will also be extracted and saved. Note that AD-AutoGPT automatically uses the given prompt and formalizes a data collection and processing pipeline based on the toolsets in our instruction library without any human intervention.

7.5.2 Spatiotemporal Information Extraction and Visualization



(a) Places where the latest news about Alzheimer's diseases happened.

(b) The number of news collected for each month from June 2022 to May 2023.

Figure 7.3: The visualization of the results from the spatial and temporal information extraction. (a) shows the spatial distribution of the Alzheimer's disease news. The news mainly happened in America and Western Europe. (b) shows the temporal change in the number of news occurrences from June 2022 to May 2023.

Based on the given prompt, AD-AutoGPT decides to use *Extract Spatial Data* tool and *Extract Temporal Data* tool in our instruction library (see Figure 7.1) to extract the places where these news articles mentioned and the timestamps when these news articles were posted online.

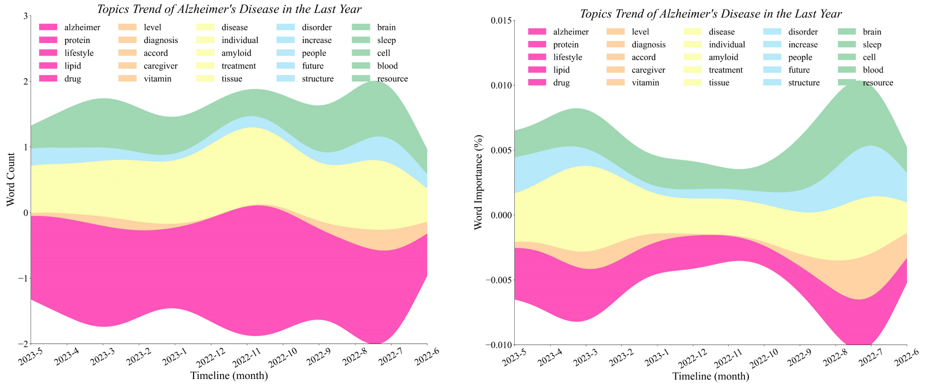
The spatial locations of extracted places from all news articles are visualized in Figure 7.3a. Note that this map visualization is automatically generated by AD-AutoGPT based on the prompt shown in Figure 7.2b. It can be seen that most of the news articles about Alzheimer's Disease in the past year mainly occurred in the United States and Western Europe. For the BBC, although it basically only reports Alzheimer's disease news in the UK, the total number of news is not inferior to that of other websites. Similarly, websites in the United States such as NIA also pay more attention to local news, especially in the southeastern states of the United States. For the Alzheimer's Association, the sources of news reports are relatively scattered all over the world, while the United States and Western Europe still show higher report frequencies than other regions such as South America, Africa, Australia, and so on. Finally, for Mayo Clinic, since there is less news from this news source, only a few occurrences can be seen on the map. Generally speaking, the distribution of news is worldwide, but it is concentrated in the southeastern United States and Western Europe. These might be because of the select bias of those four news media we use or the well-developed Alzheimer's disease research in these regions.

Temporal data analysis results can be seen in Figure 7.3b. The numbers of news reports about Alzheimer's disease in each month of the past year (June 2022 to May 2023) are visualized. It can be seen that the overall trend of the number of news reports is declining, from 31 in a single month in June 2022 to 13 in May 2023. It can also be seen that September, October, and November 2022 are the period of high incidences of news reports. The number of news reports in each of the three months exceeded 27, and those in September 2022 reached 32, which was the highest in 2022. This might be because there was news that had a profound impact on AD-related media during this period, resulting in a sudden increase in reports, which deserves special attention from users.

Therefore, it can be seen that AD-AutoGPT can not only extract useful spatiotemporal information from a wide range of news sources but can also use the visualization function to more intuitively display the spatial distribution of the AD-related news and their development through time which might be useful for users. We need to emphasize that these spatiotemporal analyses was done by AD-AutoGPT without any human input. Thereby AD-AutoGPT improves the efficiency of researchers' work, which AutoGPT cannot do because it does not design functions of information extraction from web pages.

7.5.3 LDA Topic Modeling and Hot Topic Analysis

Based on the LDA topic modeling, a hot topic analysis is automatically conducted by AD-AutoGPT. The results can be seen in Figure 7.4. AD-AutoGPT aggregates the summaries of the news reported in the past year for LDA analysis, and finally got 5 hot topics. It selects the top 5 words with the most occurrences for each of the 5 hot topics and draws streamgraphs according to the number of occurrences and word weights of the words. Please refer to Figure 7.4a and



(a) The word count trend of each topic obtained from the LDA results.

(b) The word importance trend of each topic obtained from the LDA results.

Figure 7.4: For each Topic, the Streamplot graph displays the occurrence times and frequency of different keywords in different time periods.

7.4b. In this way, you can see the changes in topic distributions according to time, so as to quickly understand the trend of the research topic.

It can be found that the keywords of the first hot topic are mainly protein, lipid, and drug, and this type of topic has occupied the largest weight in the past year, which shows that scientists are mostly concerned about seeking reliable drug treatment for Alzheimer's disease. The keywords of the topic with the second highest proportion are individual, treatment, amyloid, and tissue. This topic is also about the drug treatment of Alzheimer's disease, but the focus has obviously shifted from the research and development of new drugs to the current personal medication, reflecting the patients' concerns about self-care. The keywords of the third-ranked topic include sleep, brain, blood, cell, etc. This type of news mainly focuses on the causes of Alzheimer's disease, which is similar to popular science news. It can be seen that journalists have attached great importance to popular science in the past year. For the fourth-ranked topic, the keywords are increase, future, disorder, future, etc.

This topic is mostly related to the future plan or expectation for Alzheimer's disease research. The keywords of the last topic are mainly diagnosis, caregiver, vitamin, etc., reflecting the public's concerns about the diagnosis, care and prevention of Alzheimer's disease.

Therefore, we can conclude that through hot topic analysis, we can easily get the popular topics in the news during June 2022 - May 2023 period by using AD-AutoGPT's autonomous workflow. Users no longer need to read extensively on news, but they can easily use the help of AD-AutoGPT to understand the hot topics of Alzheimer's disease in the past period so that the efficiency of work and research on Alzheimer's disease is greatly improved. Owing to GPT-4's powerful summarizing ability, in the future, the work of early information collection can be completely handed over to AI. Humans only need to judge and focus on the most critical information returned by AI to quickly understand the development and changes in the public health domain, thus saving time and resources.

7.6 Discussion and Conclusion

7.6.1 Automating Data Analytics

The success of AD-AutoGPT shows the transformative potential of LLMs in the public health domain. By harnessing the advanced linguistic understanding and autonomous operations of AD-AutoGPT, we were able to streamline the analytical process and conduct comprehensive analyses of extensive news sources related to Alzheimer's Disease (AD). Moreover, AD-AutoGPT has the potential to go beyond the public health domain and be applied in various disciplines.

One of the key advantages of autonomous LLM-based tools such as Auto-GPT and AD-AutoGPT is their ability to automate and optimize complex data extraction and analysis tasks, as well as transcending traditional labor-intensive methods. This enables researchers and professionals across different fields to access and engage with large language models, empowering them to conduct sophisticated analyses efficiently, regardless of their technical expertise.

7.6.2 Prioritizing Insights and Innovation

Through the development of AD-AutoGPT, we conduct a detailed trend analysis, intertopic distance mapping, and identified salient terms relevant to AD. These findings provide valuable insights into the shifting focus and narrative surrounding AD, not only in the domain of public health but also in broader contexts. By quantifying and visualizing the discourse, we gain a nuanced understanding of the prevalent topics, concerns, and perspectives related to AD, facilitating targeted interventions, communication strategies, and decision-making across multiple fields.

The integration of AutoGPT and other autonomous LLM-based tools into research across different disciplines represents a significant advancement. By automating data analysis tasks, researchers can dedicate more time and resources to interpreting the results and deriving actionable insights. This accelerates the research process and enhances the accuracy and reliability of the findings in diverse areas, such as social sciences, economics, technology, and more.

7.6.3 Transforming Public Health

Furthermore, the insights obtained from this research have broader implications beyond public health. The automation capabilities of AD-AutoGPT can revolutionize the field of infodemiology by efficiently analyzing online information trends, tracking the dissemination of information and misinformation, and predicting future trends. This has the potential to inform evidence-based interventions, enhance communication strategies, and combat misinformation across various domains.

While our AD-AutoGPT has made significant strides in utilizing autonomous LLM-based tools for AD analysis in the public health domain, there are still areas for further exploration and improvement. For example, based on different underlying pathologies, AD-related dementias (ADRD) can be categorized as four major types: prion disease, AD, frontotemporal lobar degeneration (FTLD), and Lewy body diseases (LBD). In practical clinical settings, differentiations among these subtypes of dementias are very challenging, due to both mixed pathologies and clinical symptoms. Our proposed AD-AutoGPT is a general framework and can be easily extended and refined to adapt to other dementias and various brain disorders. Future studies could also focus on expanding the dataset to include a broader range of sources and different languages to capture a more comprehensive understanding of the global discourse on different dementias across different fields. Additionally, exploring the integration of AD-AutoGPT with other data sources, such as social media platforms and electronic records, could provide a more holistic perspective on ADRD conversations and outcomes across multiple disciplines.

7.6.4 Ethical Issues related to Autonomous LLM-based Tools

In the use of autonomous LLM-based tools, several ethical issues arise that warrant careful consideration. First, these models generate output based on their training data, which if biased or discriminatory, could result in outputs that perpetuate such biases (Ferrara, 2023; Magee et al., 2021). Ethical considerations must therefore include the selection and handling of training data of LLMs to minimize the risk of biased or inappropriate outputs.

In addition, issues of privacy and consent are paramount, particularly when dealing with sensitive data such as health information (Z. Liu et al., 2023). Even though LLMs do not remember specific inputs or retain personal data, the potential misuse of these tools can lead to leaking private or sensitive information , which raises significant ethical and legal questions.

Moreover, the potential for misuse extends to the propagation of false information or misinformation (Hazell, 2023; Liao, Liu, Dai, Xu, et al., 2023), a concern that is especially salient in the context of public health. LLMs can generate plausible-sounding but factually incorrect or misleading information (Latif et al., 2023; Y. Liu et al., 2023), which, if not properly managed, could have severe consequences.

Finally, the democratization of powerful technologies like AutoGPT also raises questions about responsibility and oversight. As these tools become more accessible and widespread, ensuring appropriate use and managing the potential for misuse becomes increasingly challenging.

Addressing these ethical issues is essential for the responsible development and deployment of autonomous LLM-based tools. This includes the development of robust guidelines for data handling, the implementation of safeguards against misuse, the provision of clear user instructions and warnings about potential pitfalls, and ongoing efforts to refine and improve these tools in light of user feedback and societal needs. The goal should be to harness the potential of these technologies while mitigating risks and adverse impacts, striking a balance between technology innovation and ethical responsibility.

Chapter 8

Conclusion and Future Works

8.1 Conclusion

The conclusion of my dissertation summarizes the significant research I conducted throughout my doctoral study.

First, drawing inspiration from the architecture of the brain, my research aimed to optimize neural architecture within the DARTS RNN space. The evaluation on spatial-temporal functional brain networks yielded promising results, demonstrating the potential of this approach. Furthermore, we applied the same method to PAE datasets to identify biomarkers of PAE, both at the group-wise and individual levels. The experimental results have suggested that our method can detect the temporal and parietal networks across three groups, and these networks are affected by an increase in PAE severity.

Incorporating the hierarchical structure of functional brain networks, we integrated hierarchical topology information into batch normalization within CNN models. This enhancement allowed for a better understanding of how deep neural networks process data, improving model interpretability and disentangling semantic concepts without compromising classification performance.

Building upon the insights gained from the GyriNet, we developed the novel model twin-transformer, which revealed the core roles of gyri nodes and the peripheral roles of sulci nodes. Leveraging this knowledge, we applied the core-periphery relationship pattern to the computational graph between patches in the Vision transformer, achieving state-of-the-art performance.

Continuing our investigation, we utilized causal inference approaches to explore the role of specific regions of interest (ROIs) in relation to Alzheimer's disease (AD). This line of research has led to the identification of key ROIs associated with AD, providing valuable insights into the mechanisms of the disease.

Inspired by how humans describe objects, we turned our attention to large language models, particularly ChatGPT. We discovered that ChatGPT exhibited similar capabilities to our own brains, expressing sentences in different ways while preserving semantic meaning. Leveraging ChatGPT in few-shot taxonomy tasks resulted in impressive performance.

Finally, we aimed to construct an autonomous AI-driven system (AD-AutoGPT) capable of conducting data collection, processing, and analysis autonomously in response to users' textual prompts. This system represents a significant step towards creating a self-sufficient framework for data-driven investigations.

8.2 Future Work

This dissertation represents an initial exploration into brain-inspired artificial intelligence. Indeed, there are numerous promising avenues for future research that extend beyond the existing body of work. The core-periphery pattern discovered in our brain serves as a starting point, but it's crucial to acknowledge the intricate complexity of brain function. Future investigations can delve deeper into the patterns of the brain and refine deep learning models accordingly. Conversely, leveraging advanced deep learning models can provide valuable insights into understanding the mechanisms of our brain.

The emergence of large language models, such as ChatGPT, has showcased remarkable abilities to generate human-like content. Similarly, the SAM vision model has demonstrated its potential in understanding object boundaries without the need for extensive training. Future research can explore the fusion of multiple media sources, including sound, text, audio, and images, to transform the input of deep learning models from single data types to multi-modality data. This mimics how our brain comprehends the world, utilizing our senses of sight, hearing, and cognitive understanding. The ultimate goal may lie in a unified model capable of addressing any task, thereby eliminating the need for specialized models for different modalities.

By integrating these future research directions, we can advance the field of brain-inspired artificial intelligence. The synergy between brain-inspired principles and cutting-edge deep learning models holds tremendous potential in creating more intelligent and comprehensive systems. This would enable machines to approach human-like understanding and cognitive capabilities across various domains. The journey towards achieving this goal requires continuous exploration, collaboration, and innovation in the field of brain-inspired artificial intelligence.

Bibliography

Agathokleous, E., Saitanis, C. J., Fang, C., & Yu, Z. (2023). Use of chatgpt: What does it mean for biology and environmental science? *Science of The Total Environment*, 164154.

Ahlers, D. (2013). Assessment of the accuracy of geonames gazetteer data. *Proceedings of the 7th workshop on geographic information retrieval*, 74–81.

Alexander, B., Loh, W. Y., Matthews, L. G., Murray, A. L., Adamson, C., Beare, R., Chen, J., Kelly, C. E., Anderson, P. J., Doyle, L. W., et al. (2019). Desikan-killiany-tourville atlas compatible version of m-crib neonatal parcellated whole brain atlas: The m-crib 2.0. *Frontiers in Neuroscience*, *13*, 34.

Alvarez-Hamelin, J. I., Dall'Asta, L., Barrat, A., & Vespignani, A. (2005). K-core decomposition of internet graphs: Hierarchies, self-similarity and measurement biases. *arXiv preprint cs/0511007*.

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., & Chang, K.-W. (2018). Generating Natural Language Adversarial Examples. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2890–2896. https://doi.org/10.18653/v1/D18-1316

An, L., Zhang, P., Adeli, E., Wang, Y., Ma, G., Shi, F., Lalush, D. S., Lin, W., & Shen, D. (2016). Multi-level canonical correlation analysis for standard-dose pet image estimation. *IEEE Transactions on Image Processing*, *25*(7), 3303–3315. https://doi.org/10.1109/TIP.2016.2567072

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. (2023). Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Antaki, F., Touma, S., Milad, D., El-Khoury, J., & Duval, R. (2023). Evaluating the performance of chatgpt in ophthalmology: An analysis of its successes and shortcomings. *medRxiv*, 2023–01.

Antoniou, A., Edwards, H., & Storkey, A. (2018). How to train your maml. *arXiv preprint arXiv:1810.09502*.

Archibald, S. L., Fennema-Notestine, C., Gamst, A., Riley, E. P., Mattson, S. N., & Jernigan, T. L. (2001). Brain dysmorphology in individuals with severe prenatal alcohol exposure. *Developmental medicine and child neurology*, *43*(3), 148–154.

Avramopoulos, D. (2009). Genetics of alzheimer's disease: Recent advances. *Genome medicine*, *1*(3), 1–7.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*.

Bacsu, J.-D., Cammer, A., Ahmadi, S., Azizi, M., Grewal, K. S., Green, S., Gowda-Sookochoff, R., Berger, C., Knight, S., Spiteri, R. J., et al. (2022). Examining the twitter discourse on dementia during alzheimer's awareness month in canada: Infodemiology study. *JMIR Formative Research*, *6*(10), e40049.

Bacsu, J.-D., Fraser, S., Chasteen, A. L., Cammer, A., Grewal, K. S., Bechard, L. E., Bethell, J., Green, S., McGilton, K. S., Morgan, D., et al. (2022). Using twitter to examine stigma against people with dementia during covid-19: Infodemiology study. *JMIR aging*, *5*(1), e35677.

Baidoo-Anu, D., & Owusu Ansah, L. (2023). Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning. *Available at SSRN 4337484*.

Bandoli, G., Jones, K., Wertelecki, W., Yevtushok, L., Zymak-Zakutnya, N., Granovska, I., Plotka, L., Chambers, C., & CIFASD. (2020). Patterns of prenatal alcohol exposure and alcohol-related dysmorphic features. *Alcoholism: Clinical and Experimental Research*, *44*(10), 2045–2052.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., et al. (2023). A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Banker, L., & Tadi, P. (2019). Neuroanatomy, precentral gyrus.

Bao, Y., Wu, M., Chang, S., & Barzilay, R. (2019). Few-shot text classification with distributional signatures. *arXiv preprint arXiv:1908.06039*.

Barberá, P., Wang, N., Bonneau, R., Jost, J. T., Nagler, J., Tucker, J., & González-Bailón, S. (2015). The critical periphery in the growth of social protests. *PloS one*, *10*(11), e0143611.

Barch, D. M., Burgess, G. C., Harms, M. P., Petersen, S. E., Schlaggar, B. L., Corbetta, M., Glasser, M. F., Curtiss, S., Dixit, S., Feldt, C., et al. (2013). Function in the human connectome: Task-fmri and individual differences in behavior. *Neuroimage*, *80*, 169–189.

Bassett, D. S., Wymbs, N. F., Rombach, M. P., Porter, M. A., Mucha, P. J., & Grafton, S. T. (2013). Task-based core-periphery organization of human brain dynamics. *PLoS computational biology*, *9*(9), e1003171.

Bau, A., Belinkov, Y., Sajjad, H., Durrani, N., Dalvi, F., & Glass, J. (2018). Identifying and controlling important neurons in neural machine translation. *International Conference on Learning Representations*.

Bayer, M., Kaufhold, M.-A., & Reuter, C. (2022). A survey on data augmentation for text classification. *ACM Computing Surveys*, *55*(7), 1–39.

Belinkov, Y., & Bisk, Y. (2017). Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Berant, J., Chou, A., Frostig, R., & Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1533–1544.

Bica, I., Jordon, J., & van der Schaar, M. (2020). Estimating the effects of continuous-valued interventions using generative adversarial networks. *Advances in Neural Information Processing Systems*, *33*, 16434–16445.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Blum, A. (2022). Breaking chatgpt with dangerous questions understanding how chatgpt prioritizes safety, context, and obedience.

Borgatti, S. P., & Everett, M. G. (2000). Models of core/periphery structures. *Social networks*, *21*(4), 375–395.

Boyd, J. P., Fitzgerald, W. J., & Beck, R. J. (2006). Computing core/periphery structures and permutation tests for social relations data. *Social networks*, *28*(2), 165–178.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Cai, X., Liu, S., Han, J., Yang, L., Liu, Z., & Liu, T. (2021). Chestxraybert: A pretrained language model for chest radiology report summarization. *IEEE Transactions on Multimedia*, 1–1. https://doi.org/10.1109/TMM.2021.3132724

Cai, X., Liu, S., Yang, L., Lu, Y., Zhao, J., Shen, D., & Liu, T. (2022). Covidsum: A linguistically enriched scibert-based summarization model for covid-19 scientific papers. *Journal of Biomedical Informatics*, *127*, 103999. https://doi.org/https://doi.org/10.1016/j.jbi.2022.103999

Calhoun, V. D., Eichele, T., & Pearlson, G. (2009). Functional brain networks in schizophrenia: A review. *Frontiers in human neuroscience*, 17.

Camus, V., Payoux, P., Barré, L., Desgranges, B., Voisin, T., Tauber, C., Joie, R. L., Tafani, M., Hommet, C., & Chételat, G. (2012). Using pet with 18f-av-45 (florbetapir) to quantify brain amyloid load in a clinical environment. *European Journal of Nuclear Medicine Molecular Imaging*, *39*(4), 621–631.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. *European conference on computer vision*, 213–229.

Carmi, S., Havlin, S., Kirkpatrick, S., Shavitt, Y., & Shir, E. (2007). A model of internet topology using k-shell decomposition. *Proceedings of the National Academy of Sciences*, *104*(27), 11150–11154.

Castelvecchi, D. (2022). Are chatgpt and alphacode going to replace programmers? *Nature*.

Chang, C.-H., Adam, G. A., & Goldenberg, A. (2021). Towards robust classification model by counterfactual and invariant data generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Chen, C., Li, O., Tao, C., Barnett, A. J., Su, J., & Rudin, C. (2018). This looks like that: Deep learning for interpretable image recognition. *arXiv preprint arXiv:1806.10574*.

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., & Gao, W. (2021). Pre-trained image processing transformer. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12299–12310.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., & Sutskever, I. (2020). Generative pretraining from pixels. *International Conference on Machine Learning*, 1691–1703.

Chen, M., Peng, H., Fu, J., & Ling, H. (2021). Autoformer: Searching transformers for visual recognition. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12270–12280.

Chen, R. T., Li, X., Grosse, R., & Duvenaud, D. (2019). Isolating sources of disentanglement in vaes. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2615–2625.

Chen, T., Cheng, Y., Gan, Z., Yuan, L., Zhang, L., & Wang, Z. (2021). Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, *34*, 19974–19988.

Chen, X., Hsieh, C.-J., & Gong, B. (2021). When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*.

Chen, Z., Bei, Y., & Rudin, C. (2020). Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, *2*(12), 772–782.

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, *4*(1), 266–298.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, *1*, 539–546.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Chu, Z., Rathbun, S. L., & Li, S. (2020). Matching in selective and balanced representation space for treatment effects estimation. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 205–214.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., & Batra, D. (2015). Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*.

Coulombe, C. (2018). Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs. https://doi.org/10.48550/arXiv.1812.04718
Comment: 33 pages, 25 figures

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Cucuringu, M., Rombach, P., Lee, S. H., & Porter, M. A. (2016). Detection of core–periphery structure in networks using spectral methods and geodesic paths. *European Journal of Applied Mathematics*, *27*(6), 846–887.

Dai, H., Li, Q., Zhao, L., Pan, L., Shi, C., Liu, Z., Wu, Z., Zhang, L., Zhao, S., Wu, X., et al. (2022). Graph representation neural architecture search for optimal spatial/temporal functional brain network decomposition. *Machine Learning in Medical Imaging: 13th International Workshop, MLMI 2022, Held in Conjunction with MICCAI 2022, Singapore, September 18, 2022, Proceedings*, 279–287.

Dartigues, J. F. (2009). Alzheimer's disease: A global challenge for the 21st century. *The Lancet Neurology*, *8*(12), 1082–1083.

Dave, T., Athaluri, S. A., & Singh, S. (2023). Chatgpt in medicine: An overview of its applications, advantages, limitations, future prospects, and ethical considerations. *Frontiers in Artificial Intelligence*, *6*.

De Clercq, M., Vats, A., & Biel, A. (2018). Agriculture 4.0: The future of farming technology. *Proceedings of the World Government Summit, Dubai, UAE*, 11–13.

De Juan Romero, C., & Borrell, V. (2015). Coevolution of radial glial cells and the cerebral cortex. *Glia*, *63*(8), 1303–1319.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, *29*.

Deng, F., Jiang, X., Zhu, D., Zhang, T., Li, K., Guo, L., & Liu, T. (2014). A functional model of cortical gyri and sulci. *Brain structure and function*, *219*(4), 1473–1491.

Desjardins, G., Simonyan, K., Pascanu, R., & Kavukcuoglu, K. (2015). Natural neural networks. *arXiv preprint arXiv:1507.00210*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019a). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019b). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171–4186.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, *26*(3), 297–302.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dowling, M., & Lucey, B. (2023). Chatgpt for (finance) research: The bananarama conjecture. *Finance Research Letters*, 103662.

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*.

Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. *The Journal of Machine Learning Research*, *20*(1), 1997–2017.

Erdos, P., Rényi, A., et al. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, *5*(1), 17–60.

Eysenbach, G. (2002). Infodemiology: The epidemiology of (mis) information. *The American journal of medicine*, *113*(9), 763–765.

Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, *28*(4), 594–611.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Ferrara, E. (2023). Should chatgpt be biased? challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738*.

Fezari, M., & Ali-Al-Dahoud, A. A.-D. (n.d.). From gpt to autogpt: A brief attention in nlp processing using dl.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *International conference on machine learning*, 1126–1135.

Fischl, B., & Dale, A. M. (2000). Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the National Academy of Sciences*, *97*(20), 11050–11055.

Fischl, B., Rajendran, N., Busa, E., Augustinack, J., Hinds, O., Yeo, B. T., Mohlberg, H., Amunts, K., & Zilles, K. (2008). Cortical folding patterns and predicting cytoarchitecture. *Cerebral cortex*, *18*(8), 1973–1980.

Freund, H.-J. (2002). Mechanisms of voluntary movements. *Parkinsonism & related disorders*, *9*(1), 55–59.

Frieder, S., Pinchetti, L., Griffiths, R.-R., Salvatori, T., Lukasiewicz, T., Petersen, P. C., Chevalier, A., & Berner, J. (2023). Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*.

Gallagher, R. J., Young, J.-G., & Welles, B. F. (2021). A clarified typology of core-periphery structure in networks. *Science advances*, *7*(12), eabc9800.

Gangal, V., Feng, S. Y., Alikhani, M., Mitamura, T., & Hovy, E. (2022). Nareor: The narrative reordering problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(10), 10645–10653.

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 3816–3830.

Ge, Q., Li, P., Yan, S., Lu, J., Wang, X., Zhou, Y., Paranjpe, M., Li, Y., Ng, Y. L., & Gu, F. (2022). Tracer-specific reference tissues selection improves detection of 18f-fdg, 18f-florbetapir, and 18f-flortaucipir pet suvr changes in alzheimer's disease. *Human Brain Mapping*, *43*(7), 2121–2133.

Ge, Y., Guo, Y., Yang, Y.-C., Al-Garadi, M. A., & Sarker, A. (2022). Few-shot learning for medical text: A systematic review. *arXiv preprint arXiv:2204.14081*.

Gertz, C. C., & Kriegstein, A. R. (2015). Neuronal migration dynamics in the developing ferret cortex. *Journal of Neuroscience*, *35*(42), 14307–14315.

Ghorbani, A., Abid, A., & Zou, J. (2019). Interpretation of neural networks is fragile. *AAAI*.

Ghorbani, A., Wexler, J., Zou, J. Y., & Kim, B. (2019). Towards automatic concept-based explanations. *NeurIPS*.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. *International conference on machine learning*, 1263–1272.

GitHub - geopandas/geopandas: Python tools for geographic data — github.com [[Accessed 15-Jun-2023]]. (n.d.).

GitHub - geopy/geopy: Geocoding library for Python. — github.com [[Accessed 15-Jun-2023]]. (n.d.).

Glaese, A., McAleese, N., Trębacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., et al. (2022). Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.

Glass, T. A., Goodman, S. N., Hernán, M. A., & Samet, J. M. (2013). Causal inference in public health. *Annual review of public health*, *34*, 61–75.

Glymour, M. M., & Spiegelman, D. (2017). Evaluating public health interventions: 5. causal inference in public health research—do sex, race, and biological factors cause health outcomes? *American journal of public health*, *107*(1), 81–85.

Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, *2*, 729–734.

Griffin, G., Holub, A., & Perona, P. (2007). Caltech-256 object category dataset.

Gritta, M., Pilehvar, M. T., & Collier, N. (2018). Which melbourne? augmenting geocoding with maps.

Gu, S., Xia, C. H., Ciric, R., Moore, T. M., Gur, R. C., Gur, R. E., Satterthwaite, T. D., & Bassett, D. S. (2020). Unifying the notions of modularity and core–periphery structure in functional brain networks during youth. *Cerebral Cortex*, *30*(3), 1087–1102.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., & Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, *3*(1), 1–23.

Guillon, J., Chavez, M., Battiston, F., Attal, Y., La Corte, V., Thiebaut de Schotten, M., Dubois, B., Schwartz, D., Colliot, O., & de Vico Fallani, F. (2019). Disrupted core-periphery structure of multimodal brain networks in alzheimer's disease. *Network Neuroscience*, *3*(2), 635–652.

Hair Jr, J. F., & Sarstedt, M. (2021). Data, measurement, and causal inferences in machine learning: Opportunities and challenges for marketing. *Journal of Marketing Theory and Practice*, *29*(1), 65–77.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, *30*.

Han, X., Zhao, W., Ding, N., Liu, Z., & Sun, M. (2022). Ptr: Prompt tuning with rules for text classification. *AI Open*, *3*, 182–192.

Hansen, B. B. (2008). The prognostic analogue of the propensity score. *Biometrika*, *95*(2), 481–488.

Hassanpour, N., & Greiner, R. (2019). Counterfactual regression with importance sampling weights. *IJCAI*, 5880–5887.

Hazell, J. (2023). Large language models can be used to effectively scale spear phishing campaigns. *arXiv preprint arXiv:2305.06972*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016a). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Deep residual learning for image recognition. cvpr. 2016.

He, R., & McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *proceedings of the 25th international conference on world wide web*, 507–517.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2017). Beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*.

Hilgetag, C. C., & Barbas, H. (2005). Developmental mechanics of the primate cerebral cortex. *Anatomy and embryology*, *210*(5), 411–417.

Hinton, W. L., & Levkoff, S. (1999). Constructing alzheimer's: Narratives of lost identities, confusion and loneliness in old age. *Culture, medicine and psychiatry*, *23*, 453–475.

Hirano, K., & Imbens, G. W. (2004). The propensity score with continuous treatments. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*, *226164*, 73–84.

Holme, P. (2005). Core-periphery organization of complex networks. *Physical Review E*, *72*(4), 046111.

Honey, C. J., Thivierge, J.-P., & Sporns, O. (2010). Can structure predict function in the human brain? *Neuroimage*, *52*(3), 766–776.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hu, X., Huang, H., Peng, B., Han, J., Liu, N., Lv, J., Guo, L., Guo, C., & Liu, T. (2018). Latent source mining in fmri via restricted boltzmann machine. *Human brain mapping*, *39*(6), 2368–2380.

Huang, H., Hu, X., Zhao, Y., Makkie, M., Dong, Q., Zhao, S., Guo, L., & Liu, T. (2017). Modeling task fmri data via deep convolutional autoencoder. *IEEE transactions on medical imaging*, *37*(7), 1551–1561.

Huang, H., Zhao, L., Hu, X., Dai, H., Zhang, L., Zhu, D., & Liu, T. (2022). Bi avan: Brain inspired adversarial visual attention network. *arXiv preprint arXiv:2210.15790*.

Huang, L., Yang, D., Lang, B., & Deng, J. (2018a). Decorrelated batch normalization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 791–800.

Huang, L., Yang, D., Lang, B., & Deng, J. (2018b). Decorrelated batch normalization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 791–800.

Huang, L.-K., Huang, J., Rong, Y., Yang, Q., & Wei, Y. (2022). Frustratingly easy transferability estimation. *International Conference on Machine Learning*, 9201–9225.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, *160*(1), 106.

Ibrokhimov, B., & Kang, J.-Y. (2022). Two-stage deep learning method for breast cancer detection using high-resolution mammogram images. *Applied Sciences*, *12*(9), 4616.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, 448–456.

Jabotinsky, H. Y., & Sarel, R. (2022). Co-authoring with an ai? ethical dilemmas and artificial intelligence. *Ethical Dilemmas and Artificial Intelligence (December 15, 2022).*

Jack, C. R., Wiste, H. J., Knopman, D. S., Vemuri, P., Mielke, M. M., Weigand, S. D., Senjem, M. L., Gunter, J. L., Lowe, V., & Gregg, B. E. (2014).

Rates of -amyloid accumulation are independent of hippocampal neurodegeneration. *Neurology*, *82*(18), 1605.

Jenkinson, M., Bannister, P., Brady, M., & Smith, S. (2002). Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, (2), 17.

Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., & Smith, S. M. (2012). Fsl. *Neuroimage*, *62*(2), 782–790.

Jiang, X., Zhang, T., Zhang, S., Kendrick, K. M., & Liu, T. (2021). Fundamental functional differences between gyri and sulci: Implications for brain function, cognition, and behavior. *Psychoradiology*, *1*(1), 23–41.

Jiao, W., Wang, W., Huang, J.-t., Wang, X., & Tu, Z. (2023). Is chatgpt a good translator? a preliminary study. *arXiv preprint arXiv:2301.08745*.

Jindal, A., Chowdhury, A. G., Didolkar, A., Jin, D., Sawhney, R., & Shah, R. (2020). Augmenting nlp models using latent feature interpolations. *Proceedings of the 28th International Conference on Computational Linguistics*, 6931–6936.

Johansson, F., Shalit, U., & Sontag, D. (2016). Learning representations for counterfactual inference. *International conference on machine learning*, 3020–3029.

Johnson, M. B., Wang, P. P., Atabay, K. D., Murphy, E. A., Doan, R. N., Hecht, J. L., & Walsh, C. A. (2015). Single-cell analysis reveals transcriptional heterogeneity of neural progenitors in human cortex. *Nature neuroscience*, *18*(5), 637–646.

Jones, K., & Smith, D. (1973). Recognition of the fetal alcohol syndrome in early infancy. *The Lancet*, *302*(7836), 999–1001.

Ju, Y., Adams, B., Janowicz, K., Hu, Y., Yan, B., & McKenzie, G. (2016). Things and strings: Improving place name disambiguation from short texts by combining entity co-occurrence with topic modeling. *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, 353–367.

Kalyan, K. S., Rajasekharan, A., & Sangeetha, S. (2021). Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*.

Karimzadeh, M., Pezanowski, S., MacEachren, A. M., & Wallgrün, J. O. (2019). Geotxt: A scalable geoparsing system for unstructured text geolocation. *Transactions in GIS*, *23*(1), 118–136.

Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., et al.

(2023). Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, *103*, 102274.

Kawamoto, T., Tsubaki, M., & Obuchi, T. (2018). Mean-field theory of graph neural networks in graph partitioning. *Advances in Neural Information Processing Systems*, *31*.

Kellmeyer, P. (2019). Artificial intelligence in basic and clinical neuroscience: Opportunities and ethical challenges. *Neuroforum*, *25*(4), 241–250.

Khalil, M., & Er, E. (2023). Will chatgpt get you caught? rethinking of plagiarism detection. *arXiv preprint arXiv:2302.04335*.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *ICML*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., & Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature physics*, *6*(11), 888–893.

Kobayashi, S. (2018). Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 452–457. https://doi.org/10.18653/v1/N18-2072

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., & Liang, P. (2020). Concept bottleneck models. *International Conference on Machine Learning*, 5338–5348.

Kosinski, M. (2023). Theory of mind may have spontaneously emerged in large language models. *arXiv preprint arXiv:2302.02083*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

Kumar, V., Choudhary, A., & Cho, E. (2020). Data Augmentation Using Pre-trained Transformer Models. *arXiv preprint arXiv:2003.02245*.

Kumar, V., Glaude, H., de Lichy, C., & Campbell, W. (2019). A closer look at feature space data augmentation for few-shot intent classification.

Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019), 1–10.

Kung, T. H., Cheatham, M., Medenilla, A., Sillos, C., De Leon, L., Elepaño, C., Madriaga, M., Aggabao, R., Diaz-Candido, G., Maningo, J., et al. (2023). Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models. *PLOS Digital Health*, *2*(2), e0000198.

Landau, S. M., Thomas, B. A., Thurfjell, L., Schmidt, M., Margolin, R., Mintun, M., Pontecorvo, M., Baker, S. L., Jagust, W. J., & the Alzheimer's Disease Neuroimaging Initiative. (2014). Amyloid pet imaging in alzheimer's disease: A comparison of three radiotracers. *European Journal of Nuclear Medicine and Molecular Imaging*, *41*, 1398–1407.

Latif, E., Mai, G., Nyaaba, M., Wu, X., Liu, N., Lu, G., Li, S., Liu, T., & Zhai, X. (2023). Artificial general intelligence (agi) for education. *arXiv preprint arXiv:2304.12479*.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), 1995.

Lee, H.-y., Li, S.-W., & Vu, N. T. (2022). Meta learning for natural language processing: A survey. *arXiv preprint arXiv:2205.01500*.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., & Li, L. (2020). On the sentence embeddings from pre-trained language models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9119–9130.

Li, G., Liu, T., Ni, D., Lin, W., Gilmore, J. H., & Shen, D. (2015). Spatiotemporal patterns of cortical fiber density in developing infants, and their relationship with cortical thickness. *Human brain mapping*, *36*(12), 5183–5195.

Li, O., Liu, H., Chen, C., & Rudin, C. (2018). Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1).

Li, Q., Dong, Q., Ge, F., Qiang, N., Zhao, Y., Wang, H., Huang, H., Wu, X., & Liu, T. (2019). Simultaneous spatial-temporal decomposition of connectome-scale brain networks by deep sparse recurrent auto-encoders. *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26*, 579–591.

Li, Q., Wu, X., & Liu, T. (2021). Differentiable neural architecture search for optimal spatial/temporal brain function network decomposition. *Medical Image Analysis*, *69*, 101974.

Li, Q., Wu, X., Xie, F., Chen, K., Yao, L., Zhang, J., Guo, X., Li, R., & the Alzheimer's Disease Neuroimaging Initiative. (2018). Aberrant connectivity in mild cognitive impairment and alzheimer disease revealed by multimodal neuroimaging data. *Neurodegenerative Diseases*, *18*, 5–18.

Li, Q., Wu, X., Xu, L., Chen, K., Yao, L., & Li, R. (2017). Multi-modal discriminative dictionary learning for alzheimer's disease and mild cognitive impairment. *Computer methods and programs in biomedicine*, *150*, 1–8.

Li, Q., Zhang, W., Lv, J., Wu, X., & Liu, T. (2020). Neural architecture search for optimization of spatial-temporal brain network decomposition. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23*, 377–386.

Liang, C., Berant, J., Le, Q., Forbus, K., & Lao, N. (2017). Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 23–33.

Liao, W., Liu, Z., Dai, H., Wu, Z., Zhang, Y., Huang, X., Chen, Y., Jiang, X., Zhu, D., Liu, T., Li, S., Li, X., & Cai, H. (2023). Mask-guided bert for few shot text classification. *arXiv preprint arXiv:2302.10447*.

Liao, W., Liu, Z., Dai, H., Xu, S., Wu, Z., Zhang, Y., Huang, X., Zhu, D., Cai, H., Liu, T., et al. (2023). Differentiate chatgpt-generated and human-written medical texts. *arXiv preprint arXiv:2304.11567*.

Liu, H., Simonyan, K., & Yang, Y. (2018). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

Liu, H., Zhang, S., Jiang, X., Zhang, T., Huang, H., Ge, F., Zhao, L., Li, X., Hu, X., Han, J., et al. (2019a). The cerebral cortex is bisectionally segregated into two fundamentally different functional units of gyri and sulci. *Cerebral Cortex*, *29*(10), 4238–4252.

Liu, H., Zhang, S., Jiang, X., Zhang, T., Huang, H., Ge, F., Zhao, L., Li, X., Hu, X., Han, J., et al. (2019b). The cerebral cortex is bisectionally segregated into two fundamentally different functional units of gyri and sulci. *Cerebral Cortex*, *29*(10), 4238–4252.

Liu, S., Ge, F., Zhao, L., Wang, T., Ni, D., & Liu, T. (2022). Nas-optimized topology-preserving transfer learning for differentiating cortical folding patterns. *Medical Image Analysis*, *77*, 102316.

Liu, X., Zhou, M., Shi, G., Du, Y., Zhao, L., Wu, Z., Liu, D., Liu, T., & Hu, X. (2023). Coupling artificial neurons in bert and biological neurons in the human brain. *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI*.

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., et al. (2023). Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *arXiv preprint arXiv:2304.01852*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Liu, Z., He, M., Jiang, Z., Wu, Z., Dai, H., Zhang, L., Luo, S., Han, T., Li, X., Jiang, X., et al. (2022). Survey on natural language processing in medical image analysis. *Zhong nan da xue xue bao. Yi xue ban= Journal of Central South University. Medical Sciences*, *47*(8), 981–993.

Liu, Z., Yu, X., Zhang, L., Wu, Z., Cao, C., Dai, H., Zhao, L., Liu, W., Shen, D., Li, Q., et al. (2023). Deid-gpt: Zero-shot medical text de-identification by gpt-4. *arXiv preprint arXiv:2303.11032*.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., et al. (2023). The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Lu, G., Li, S., Mai, G., Sun, J., Zhu, D., Chai, L., Sun, H., Wang, X., Dai, H., Liu, N., Xu, R., Petti, D., Li, C., Liu, T., et al. (2023). Agi for agriculture. *arXiv preprint arXiv:2304.06136*.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *NIPS*.

Luo, P. (2017). Learning deep architectures via generalized whitened neural networks. *International Conference on Machine Learning*, 2238–2246.

Lv, B.-M., Quan, Y., & Zhang, H.-Y. (2021). Causal inference in microbiome medicine: Principles and applications. *Trends in microbiology*, *29*(8), 736–746.

Lv, J., Jiang, X., Li, X., Zhu, D., Chen, H., Zhang, T., Zhang, S., Hu, X., Han, J., Huang, H., et al. (2015). Sparse representation of whole-brain fmri signals for identification of functional networks. *Medical image analysis*, *20*(1), 112–134.

Lv, J., Jiang, X., Li, X., Zhu, D., Zhang, S., Zhao, S., Chen, H., Zhang, T., Hu, X., Han, J., et al. (2014). Holistic atlases of functional networks and interactions reveal reciprocal organizational architecture of cortical function. *IEEE Transactions on Biomedical Engineering*, *62*(4), 1120–1131.

Lv, J., Jiang, X., Li, X., Zhu, D., Zhao, S., Zhang, T., Hu, X., Han, J., Guo, L., Li, Z., et al. (2015). Assessing effects of prenatal alcohol exposure using group-wise sparse representation of fmri data. *Psychiatry Research: Neuroimaging*, *233*(2), 254–268.

Ma, E. (2019). Nlp augmentation.

Mackey, T., Baur, C., Eysenbach, G., et al. (2022). Advancing infodemiology in a digital intensive era. *JMIR Infodemiology*, *2*(1), e37115.

Magee, L., Ghahremanlou, L., Soldatic, K., & Robertson, S. (2021). Intersectional bias in causal language models. *arXiv preprint arXiv:2107.07691*.

Mai, G., Cundy, C., Choi, K., Hu, Y., Lao, N., & Ermon, S. (2022). Towards a foundation model for geospatial artificial intelligence (vision paper). *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 1–4.

Mai, G., Hu, Y., Gao, S., Cai, L., Martins, B., Scholz, J., Gao, J., & Janowicz, K. (2022). Symbolic and subsymbolic geoai: Geospatial knowledge graphs and spatially explicit machine learning. *Trans GIS*, *26*(8), 3118–3124.

Mai, G., Huang, W., Sun, J., Song, S., Mishra, D., Liu, N., Gao, S., Liu, T., Cong, G., Hu, Y., et al. (2023). On the opportunities and challenges of foundation models for geospatial artificial intelligence. *arXiv preprint arXiv:2304.06798*.

Mai, G., Janowicz, K., Zhu, R., Cai, L., & Lao, N. (2021). Geographic question answering: Challenges, uniqueness, classification, and future directions. *AGILE: GIScience series*, *2*, 8.

Marblestone, A. H., Wayne, G., & Kording, K. P. (2016). Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, 94.

Mattson, S. N., Bernes, G. A., & Doyle, L. R. (2019). Fetal alcohol spectrum disorders: A review of the neurobehavioral deficits associated with prenatal alcohol exposure. *Alcoholism: Clinical and Experimental Research*, *43*(6), 1046–1062.

Mavragani, A. (2020). Infodemiology and infoveillance: Scoping review. *Journal of medical internet research*, *22*(4), e16206.

McGee, R. W. (2023). Is chat gpt biased against conservatives? an empirical study. *An Empirical Study (February 15, 2023)*.

McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, *3*(29), 861.

Meilia, P. D. I., Freeman, M. D., & Zeegers, M. P. (2020). A review of causal inference in forensic medicine. *Forensic Science, Medicine and Pathology*, *16*, 313–320.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, *26*.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Miller, M. B., Huang, A. Y., Kim, J., Zhou, Z., Kirkham, S. L., Maury, E. A., Ziegenfuss, J. S., Reed, H. C., Neil, J. E., & Rento, L. a. (2022). Somatic genomic changes in single alzheimer's disease neurons. *Nature*, *604*.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heinz, I., & Roth, D. (2021). Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv preprint arXiv:2111.01243*.

Moreira, I. C., Amaral, I., Domingues, I., Cardoso, A., Cardoso, M. J., & Cardoso, J. S. (2012). Inbreast: Toward a full-field digital mammographic database. *Academic radiology*, *19*(2), 236–248.

Morgan, S. L., & Winship, C. (2015). *Counterfactuals and causal inference*. Cambridge University Press.

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., & Qi, Y. (2020). Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 119–126.

Mrkšić, N., Ó Séaghdha, D., Thomson, B., Gašić, M., Rojas-Barahona, L. M., Su, P.-H., Vandyke, D., Wen, T.-H., & Young, S. (2016). Counter-fitting Word Vectors to Linguistic Constraints. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 142–148. https://doi.org/10.18653/v1/N16-1018

Mu, J., & Andreas, J. (2020). Compositional explanations of neurons. *arXiv preprint arXiv:2006.14032*.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Interpretable machine learning: Definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*.

Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., & Edunov, S. (2020). Facebook fair's wmt19 news translation task submission. *Proc. of WMT*.

Nie, J., Guo, L., Li, K., Wang, Y., Chen, G., Li, L., Chen, H., Deng, F., Jiang, X., Zhang, T., et al. (2012). Axonal fiber terminations concentrate on gyri. *Cerebral cortex*, *22*(12), 2831–2839.

Nie, L., Ye, M., Liu, Q., & Nicolae, D. (2021). Vcnet and functional targeted regularization for learning causal effects of continuous treatments. *arXiv preprint arXiv:2103.07861*.

Niu, T., & Bansal, M. (2018). Adversarial Over-Sensitivity and Over-Stability Strategies for Dialogue Models. *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 486–496. https://doi.org/10.18653/v1/K18-1047

Norton, I., Essayed, W. I., Zhang, F., Pujol, S., Yarmarkovich, A., Golby, A. J., Kindlmann, G., Wassermann, D., Estepar, R. S. J., Rathi, Y., et al. (2017). Slicerdmri: Open source diffusion mri software for brain cancer research. *Cancer research*, *77*(21), e101–e103.

Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. *Distill*, *2*(11), e7.

OpenAI, R. (2023). Gpt-4 technical report. *arXiv*.

Ossenkoppele, R., Pichet Binette, A., Groot, C., Smith, R., Strandberg, O., Palmqvist, S., Stomrud, E., Tideman, P., Ohlsson, T., Jögi, J., Johnson, K., Sperling, R., Dore, V., Masters, C. L., Rowe, C., Visser, D., van Berckel, B. N. M., van der Flier, W. M., Baker, S., … Hansson, O. (2022). Amyloid and tau pet-positive cognitively unimpaired individuals are at high risk for future cognitive decline. *Nature Medicine*, *28*, 2381–2387.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744.

Pal, K. K., & Sudeep, K. (2016). Preprocessing for image classification by convolutional neural networks. *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 1778–1781.

Pang, T., Zhao, S., Han, J., Zhang, S., Guo, L., & Liu, T. (2022). Gumbel-softmax based neural architecture search for hierarchical brain networks decomposition. *Medical Image Analysis*, *82*, 102570.

Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B., & Callison-Burch, C. (2015). Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 425–430.

Pavlik, J. V. (2023). Collaborating with chatgpt: Considering the implications of generative artificial intelligence for journalism and media education. *Journalism & Mass Communication Educator*, 10776958221149577.

Piamonte, B. L. C., Anlacan, V. M. M., Jamora, R. D. G., & Espiritu, A. I. (2022). Googling alzheimer disease: An infodemiological and ecological study. *Dementia and Geriatric Cognitive Disorders Extra*, *11*(3), 333–339.

Post, S. G. (2000). *The moral challenge of alzheimer disease: Ethical issues from diagnosis to dying*. JHU Press.

Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., & Yang, D. (2023). Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, *21*(1), 5485–5551.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Chen, X., & Wang, X. (2021). A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, *54*(4), 1–34.

Rezayi, S., Dai, H., Liu, Z., Wu, Z., Hebbar, A., Burns, A. H., Zhao, L., Zhu, D., Li, Q., Liu, W., et al. (2022). Clinicalradiobert: Knowledge-infused few shot learning for clinical notes named entity recognition. *Machine Learning in Medical Imaging: 13th International Workshop, MLMI 2022, Held in Conjunction with MICCAI 2022, Singapore, September 18, 2022, Proceedings*, 269–278.

Rezayi, S., Liu, Z., Wu, Z., Dhakal, C., Ge, B., Zhen, C., Liu, T., & Li, S. (2022). Agribert: Knowledge-infused agricultural language models for matching food and nutrition. *International Joint Conference on Artificial Intelligence, July 23-29, 2022, Vienna, Austria*.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. *KDD*.

Rice, D. P., Fox, P. J., Max, W., Webber, P. A., Hauck, W. W., Lindeman, D. A., & Segura, E. (1993). The economic burden of alzheimer's disease care. *Health affairs*, *12*(2), 164–176.

Richards, T. B. (2023). Auto-gpt: An autonomous gpt-4 experiment.

Richiardi, J., Altmann, A., Milazzo, A.-C., Chang, C., Chakravarty, M. M., Banaschewski, T., Barker, G. J., Bokde, A. L., Bromberg, U., Büchel, C., et al. (2015). Correlated gene expression supports synchronous activity in brain networks. *Science*, *348*(6240), 1241–1244.

Rombach, M. P., Porter, M. A., Fowler, J. H., & Mucha, P. J. (2014). Core-periphery structure in networks. *SIAM Journal on Applied mathematics*, *74*(1), 167–190.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Ross, A. S., & Doshi-Velez, F. (2018). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *Thirty-second AAAI conference on artificial intelligence*.

Roth, C., Kang, S. M., Batty, M., & Barthelemy, M. (2012). A long-time limit for world subway networks. *Journal of The Royal Society Interface*, *9*(75), 2540–2550.

Rothman, K. J., & Greenland, S. (2005). Causation and causal inference in epidemiology. *American journal of public health*, *95*(S1), S144–S150.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, *66*(5), 688.

Rudin, C. (2019a). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215.

Rudin, C. (2019b). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & Aberman, K. (2022). Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *ArXiv*, *abs/1910.01108*.

Santhanam, P., Li, Z., Hu, X., Lynch, M. E., & Coles, C. D. (2009). Effects of prenatal alcohol exposure on brain activation during an arithmetic task: An fmri study. *Alcoholism: Clinical and Experimental Research*, *33*(11), 1901–1908.

Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. (2022). Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, 593–607.

Schmidhuber, J., & Blog, A. (2020). The 2010s: Our decade of deep learning/outlook on the 2020s. *The recent decade's most important developments and industrial applications based on our AI, with an outlook on the 2020s, also addressing privacy and data markets*.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schwab, P., Linhardt, L., Bauer, S., Buhmann, J. M., & Karlen, W. (2020). Learning counterfactual representations for estimating individual dose-response curves. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 5612–5619.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision*.

Sennrich, R., Haddow, B., & Birch, A. (2015). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Sennrich, R., Haddow, B., & Birch, A. (2016). Improving Neural Machine Translation Models with Monolingual Data. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 86–96. https://doi.org/10.18653/v1/P16-1009

Shalit, U., Johansson, F. D., & Sontag, D. (2017). Estimating individual treatment effect: Generalization bounds and algorithms. *International Conference on Machine Learning*, 3076–3085.

Shen, Y., Heacock, L., Elias, J., Hentel, K. D., Reig, B., Shih, G., & Moy, L. (2023). Chatgpt and other large language models are double-edged swords.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text data augmentation for deep learning. *Journal of big Data*, *8*, 1–34.

Siarohin, A., Sangineto, E., & Sebe, N. (2018). Whitening and coloring batch transform for gans. *arXiv preprint arXiv:1806.00420*.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., & Wattenberg, M. (2017). Smoothgrad: Removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Smith, S. M. (2002). Fast robust automated brain extraction.

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, *25*(1), 1.

Su, X., You, S., Xie, J., Zheng, M., Wang, F., Qian, C., Zhang, C., Wang, X., & Xu, C. (2021). Vitas: Vision transformer architecture search. *arXiv preprint arXiv:2106.13700*.

Sun, L., Xia, C., Yin, W., Liang, T., Yu, P. S., & He, L. (2020). Mixup-transformer: Dynamic data augmentation for nlp tasks. *arXiv preprint arXiv:2010.02394*.

Susnjak, T. (2022). Chatgpt: The end of online exam integrity? *arXiv preprint arXiv:2212.09292*.

Tang, Y., Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., & Tao, D. (2022). Patch slimming for efficient vision transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12165–12174.

Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, *3*(2), 334–340.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al. (2021). Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, *34*, 24261–24272.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. *International Conference on Machine Learning*, 10347–10357.

Troiani, V., Patti, M. A., & Adamson, K. (2020). The use of the orbitofrontal h-sulcus as a reference frame for value signals. *European Journal of Neuroscience*, *51*(9), 1928–1943.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K., Consortium, W.-M. H., et al. (2013). The wu-minn human connectome project: An overview. *Neuroimage*, *80*, 62–79.

van Dis, E. A., Bollen, J., Zuidema, W., van Rooij, R., & Bockting, C. L. (2023). Chatgpt: Five priorities for research. *Nature*, *614*(7947), 224–226.

Varian, H. R. (2016). Causal inference in economics and marketing. *Proceedings of the National Academy of Sciences*, *113*(27), 7310–7315.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242.

Walker, R. (1992). Implementing discrete mathematics: Combinatorics and graph theory with mathematica, steven skiena. pp 334. 1990. isbn 0-201-50943-1 (addison-wesley). *The Mathematical Gazette*, *76*(476), 286–288.

Wan, E. A., & Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, 153–158.

Wan, E. A., & Van Der Merwe, R. (2001). The unscented kalman filter. *Kalman filtering and neural networks*, 221–280.

Wang, C., Wang, J., Qiu, M., Huang, J., & Gao, M. (2021). Transprompt: Towards an automatic transferable prompting framework for few-shot text classification. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2792–2802.

Wang, C., Pan, S., Long, G., Zhu, X., & Jiang, J. (2017). Mgae: Marginalized graph autoencoder for graph clustering. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 889–898.

Wang, H., Zhao, S., Dong, Q., Cui, Y., Chen, Y., Han, J., Xie, L., & Liu, T. (2018). Recognizing brain states using deep sparse recurrent neural network. *IEEE transactions on medical imaging*, *38*(4), 1058–1068.

Wang, J., Wang, C., Luo, F., Tan, C., Qiu, M., Yang, F., Shi, Q., Huang, S., & Gao, M. (2022). Towards unified prompt tuning for few-shot text classification. *arXiv preprint arXiv:2205.05313*.

Wang, J., & Dong, Y. (2020). Measurement of text similarity: A survey. *Information*, *11*(9), 421.

Wang, J., Oh, J., Wang, H., & Wiens, J. (2018). Learning credible models. *KDD*.

Wang, J., Hu, Y., & Joseph, K. (2020). Neurotpr: A neuro-net toponym recognition model for extracting locations from social media messages. *Transactions in GIS*, *24*(3), 719–735.

Wang, L., Zhang, L., & Zhu, D. (2020). Learning latent structure over deep fusion model of mild cognitive impairment. *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 1039–1043.

Wang, R., Fu, B., Fu, G., & Wang, M. (2017). Deep & cross network for ad click predictions. In *Proceedings of the adkdd'17* (pp. 1–7).

Wang, S., Ouyang, X., Liu, T., Wang, Q., & Shen, D. (2022). Follow my eye: Using gaze to supervise computer-aided diagnosis. *IEEE Transactions on Medical Imaging*.

Wang, S., Zhao, Z., Ouyang, X., Wang, Q., & Shen, D. (2023). Chatcad: Interactive computer-aided diagnosis on medical image using large language models. *arXiv preprint arXiv:2302.07257*.

Wang, S., Liu, X., Liu, B., & Dong, D. (2022). Sentence-aware adversarial meta-learning for few-shot text classification. *Proceedings of the 29th International Conference on Computational Linguistics*, 4844–4852.

Wang, W. Y., & Yang, D. (2015). That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2557–2563.

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, *53*(3), 1–34.

Wang, Y., Huang, R., Song, S., Huang, Z., & Huang, G. (2021). Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in Neural Information Processing Systems*, *34*, 11960–11973.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of'small-world'networks. *nature*, *393*(6684), 440–442.

Wei, J., Huang, C., Vosoughi, S., Cheng, Y., & Xu, S. (2021). Few-shot text classification with triplet networks, data augmentation, and curriculum learning. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5493–5500.

Wei, J., & Zou, K. (2019a). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6382–6388. https://doi.org/10.18653/v1/D19-1670

Wei, J., & Zou, K. (2019b). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Wen, Z., & Yin, W. (2013). A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, *142*(1), 397–434.

White, C., Neiswanger, W., & Savani, Y. (2021). Bananas: Bayesian optimization with neural architectures for neural architecture search. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 10293–10301.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, 5–32.

Woolrich, M. W., Jenkinson, M., Brady, J. M., & Smith, S. M. (2004). Fully bayesian spatio-temporal modeling of fmri data. *IEEE transactions on medical imaging*, *23*(2), 213–231.

Wozniak, J. R., Mueller, B. A., Mattson, S. N., Coles, C. D., Kable, J. A., Jones, K. L., Boys, C. J., Lim, K. O., Riley, E. P., Sowell, E. R., et al. (2017). Functional connectivity abnormalities and associated cognitive deficits in fetal alcohol spectrum disorders (fasd). *Brain imaging and behavior*, *11*, 1432–1445.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., & Sun, X. (2022). Evo-vit: Slow-fast token evolution for dynamic vision transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(3), 2964–2972.

Yan, S., Zheng, Y., Ao, W., Zeng, X., & Zhang, M. (2020). Does unsupervised architecture representation learning help neural architecture search? *Advances in Neural Information Processing Systems*, *33*, 12486–12498.

Yang, S., Zhao, Z., Cui, H., Zhang, T., Zhao, L., He, Z., Liu, H., Guo, L., Liu, T., Becker, B., et al. (2019). Temporal variability of cortical gyral-sulcal resting state functional activity correlates with fluid intelligence. *Frontiers in neural circuits*, 36.

Yao, L., Li, S., Li, Y., Huai, M., Gao, J., & Zhang, A. (2018). Representation learning for treatment effect estimation from observational data. *Advances in Neural Information Processing Systems*, *31*.

Yao, L., Li, Y., Li, S., Liu, J., Huai, M., Zhang, A., & Gao, J. (2022). Concept-level model interpretation from the causal aspect. *IEEE Transactions on Knowledge and Data Engineering*.

Yao, X., Zhu, J., Huo, G., Xu, N., Liu, X., & Zhang, C. (2021). Model-agnostic multi-stage loss optimization meta learning. *International Journal of Machine Learning and Cybernetics*, *12*(8), 2349–2363.

Yazdani, A., & Boerwinkle, E. (2015). Causal inference in the age of decision medicine. *Journal of data mining in genomics & proteomics*, *6*(1).

Yin, W. (2020). Meta-learning for few-shot natural language processing: A survey. *arXiv preprint arXiv:2007.09604*.

You, J., Leskovec, J., He, K., & Xie, S. (2020). Graph structure of neural networks. *International Conference on Machine Learning*, 10881–10891.

Yu, H., & Wu, J. (2021). A unified pruning framework for vision transformers. *arXiv preprint arXiv:2111.15127*.

Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., & Yan, S. (2022). Metaformer is actually what you need for vision. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10819–10829.

Yu, X., Zhang, L., Dai, H., Lyu, Y., Zhao, L., Wu, Z., Liu, D., Liu, T., & Zhu, D. (2023). Core-periphery principle guided redesign of self-attention in transformers. *arXiv preprint arXiv:2303.15569*.

Yu, X., Zhang, L., Dai, H., Zhao, L., Lyu, Y., Wu, Z., Liu, T., & Zhu, D. (2023). Gyri vs. sulci: Disentangling brain core-periphery functional networks via twin-transformer. *arXiv preprint arXiv:2302.00146*.

Zarifhonarvar, A. (2023). Economics of chatgpt: A labor market view on the occupational impact of artificial intelligence. *Available at SSRN 4350925*.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *European conference on computer vision*, 818–833.

Zhang, F., Noh, T., Juvekar, P., Frisken, S. F., Rigolo, L., Norton, I., Kapur, T., Pujol, S., Wells III, W., Yarmarkovich, A., et al. (2020). Slicerdmri: Diffusion mri and tractography research software for brain cancer surgery planning and visualization. *JCO clinical cancer informatics*, *4*, 299–309.

Zhang, L., Wang, M., Liu, M., & Zhang, D. (2020). A survey on deep learning for neuroimaging-based brain disorder analysis. *Frontiers in neuroscience*, *14*, 779.

Zhang, L., Wang, L., & Zhu, D. (2020). Jointly analyzing alzheimer's disease related structure-function using deep cross-model attention network. *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 563–567.

Zhang, L., Wang, L., & Zhu, D. (2021). Representing alzheimer's disease progression via deep prototype tree. *arXiv preprint arXiv:2102.06847*.

Zhang, L., Zhao, L., Liu, D., Wu, Z., Wang, X., Liu, T., & Zhu, D. (2023). Cortex2vector: Anatomical embedding of cortical folding patterns. *Cerebral Cortex*, *33*(10), 5851–5862.

Zhang, M., & Chen, Y. (2018). Link prediction based on graph neural networks. *Advances in neural information processing systems*, *31*.

Zhang, S., Dong, Q., Zhang, W., Huang, H., Zhu, D., & Liu, T. (2019). Discovering hierarchical common brain networks via multimodal deep belief network. *Medical image analysis*, *54*, 238–252.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022). Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, W., Zhao, L., Li, Q., Zhao, S., Dong, Q., Jiang, X., Zhang, T., & Liu, T. (2019). Identify hierarchical structures from task-based fmri data via hybrid spatiotemporal neural architecture search net. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd*

International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22, 745–753.

Zhang, W., Zhao, S., Hu, X., Dong, Q., Huang, H., Zhang, S., Zhao, Y., Dai, H., Ge, F., Guo, L., et al. (2020). Hierarchical organization of functional brain networks revealed by hybrid spatiotemporal deep learning. *Brain connectivity*, *10*(2), 72–82.

Zhang, Y., Zhang, H., Lipton, Z. C., Li, L. E., & Xing, E. (2022). Exploring transformer backbones for heterogeneous treatment effect estimation. *NeurIPS ML Safety Workshop*.

Zhang, Y., Lyu, H., Liu, Y., Zhang, X., Wang, Y., & Luo, J. (2021). Monitoring depression trends on twitter during the covid-19 pandemic: Observational study. *JMIR infodemiology*, *1*(1), e26769.

Zhang, Y., Choi, M., Han, K., & Liu, Z. (2021). Explainable semantic space by grounding language to vision with cross-modal contrastive learning. *Advances in Neural Information Processing Systems*, *34*, 18513–18526.

Zhao, G. T. (n.d.). A comprehensive and hands-on guide to autonomous agents with gpt.

Zhao, L., Dai, H., Wu, Z., Xiao, Z., Zhang, L., Liu, D. W., Hu, X., Jiang, X., Li, S., Zhu, D., et al. (2023). Coupling visual semantics of artificial neural networks and human brain function via synchronized activations. *IEEE Transactions on Cognitive and Developmental Systems*.

Zhao, L., Wu, Z., Dai, H., Liu, Z., Zhang, T., Zhu, D., & Liu, T. (2022). Embedding human brain function via transformer. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part I*, 366–375.

Zhao, L., Zhang, L., Wu, Z., Chen, Y., Dai, H., Yu, X., Liu, Z., Zhang, T., Hu, X., Jiang, X., et al. (2023). When brain-inspired ai meets agi. *arXiv preprint arXiv:2303.15935*.

Zhao, S., Han, J., Lv, J., Jiang, X., Hu, X., Zhang, S., Lynch, M. E., Coles, C., Guo, L., Hu, X., et al. (2016). A multi-stage sparse coding framework to explore the effects of prenatal alcohol exposure. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part I 19*, 28–36.

Zhao, Y., Kuo, T.-C., Weir, S., Kramer, M. S., & Ash, A. S. (2008). Healthcare costs and utilization for medicare beneficiaries with alzheimer's. *BMC health services research*, *8*(1), 1–8.

Zhao, Y., Dai, H., Zhang, W., Ge, F., & Liu, T. (2019). Two-stage spatial temporal deep learning framework for functional brain network modeling. *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 1576–1580.

Zhao, Y., Li, X., Zhang, W., Zhao, S., Makkie, M., Zhang, M., Li, Q., & Liu, T. (2018). Modeling 4d fmri data via spatio-temporal convolutional neural networks (st-cnn). *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part III 11*, 181–189.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6881–6890.

Zhou, B., Bau, D., Oliva, A., & Torralba, A. (2018). Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., et al. (2023). A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.

Zhou, L., Zhou, Y., Corso, J. J., Socher, R., & Xiong, C. (2018). End-to-end dense video captioning with masked transformer. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8739–8748.

Zhu, M., Tang, Y., & Han, K. (2021). Vision transformer pruning. *arXiv preprint arXiv:2104.08500*.

Zhu, X., Suk, H.-I., & Shen, D. (2014). Multi-modality canonical feature selection for alzheimer's disease diagnosis. In P. Golland, N. Hata, C. Barillot, J. Hornegger, & R. Howe (Eds.), *Medical image computing and computer-assisted intervention – miccai 2014* (pp. 162–169). Springer International Publishing.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., & Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

Zielinski, C. (2021). Infodemics and infodemiology: A short history, a long future. *Revista panamericana de salud publica*, *45*, e40.

Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67*(2), 301–320.