Pattern Recognition 176 (2026) 113290

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Diagnosing Alzheimer's disease using hypergraph neural networks with prompt tuning

Chenyu Liu [a,*], Luca Cosmo [b], Luca Rossi [a,c]

[a] Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong SAR,
[b] Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice, Venice, Italy
[c] Department of Data Science and Artificial Intelligence, The Hong Kong Polytechnic University, Hung Hom, Hong Kong SAR,

ARTICLE INFO

Keywords:
Alzheimer's disease
Graph neural networks
Prompt learning

ABSTRACT

The accurate diagnosis of Alzheimer's disease (AD) and prognosis of mild cognitive impairment (MCI) conversion are crucial for early intervention. However, existing multimodal methods face several challenges, from the heterogeneity of input data, to underexplored modality interactions, missing data due to patient dropouts, and limited data caused by the time-consuming and costly data collection process. In this paper, we propose a novel Prompted Hypergraph Neural Network (PHGNN) framework that addresses these limitations by integrating hypergraph based learning with prompt learning. Hypergraphs capture higher-order relationships between different modalities, while our prompt learning approach for hypergraphs, adapted from NLP, enables efficient training with limited data. Our model is validated through extensive experiments on the ADNI dataset as well as cross-domain validations using the OASIS-3 and NACC datasets. The results demonstrate that PHGNN outperforms SOTA methods in both AD diagnosis and MCI conversion prediction, showing superior cross-domain generalization capabilities. At the same time, it uses only a fraction (6%) of the tunable parameters of traditional fine-tuning and maintains a low computational load compared to alternative tuning strategies.

1. Introduction

Alzheimer's disease (AD) is one of the most common diseases in elderly people, caused by the irreversible loss of neurons and genetically complex disorders. Therefore, accurate recognition of AD and its precursor stage, mild cognitive impairment (MCI), has attracted widespread attention, as MCI has been shown to be the optimal stage to treat in order to prevent the MCI-to-AD conversion [1]. This in turn highlights the importance of progressive MCI prediction, which aims to distinguish progressive MCI (pMCI), which may progress to AD within 36 months, from stable MCI (sMCI). In recent years, several deep learning based computer-aided diagnosis methods [2–5] have been proposed to diagnose AD and MCI using imaging and non-imaging data as different modalities that carry complementary information about the disease.

However, existing multimodal approaches for AD diagnosis suffer from three major limitations: 1) different modalities are highly heterogeneous, especially between imaging and non-imaging data; 2) patient dropouts can result in some subjects not having data from a specific modality, resulting in missing data; 3) more in general, in the field of clinical research the amount of data is often very limited, hindering the performance of deep learning models that need to be trained on large amounts of data in order to make accurate predictions.

The objective of the present study is to address these limitations. To tackle the first issue, we propose to use a hypergraph neural network (HGNN) based approach, which has already shown its effectiveness in AD diagnosis [6,7], based on the following two intuitions: 1) hypergraphs can model higher-order relationships directly by allowing a single hyperedge to connect multiple nodes, thus allowing us to better represent the rich interactions between different modalities [8,9]; 2) hypergraphs can leverage the remaining modalities by exploiting the connections between the available data points. The rationale for using many-to-many connections stems from the high heterogeneity of AD, where different combinations of biomarkers can lead to similar clinical outcomes. Consider a specific, high-risk AD subtype defined by a concurrent set of biomarkers: severe hippocampal atrophy (MRI), significant hypometabolism in the precuneus (PET), and pathological CSF p-tau levels. If Patient A, Patient B, Patient C, and Patient D all exhibit this exact combination, a hypergraph can explicitly define a single hyperedge $e = (A, B, C, D)$. This hyperedge allows the model to learn the higher-order pattern "These four patients belong to the same specific clinical subtype". A standard graph would model this as a clique of

* Corresponding author.
E-mail address: chen-yu.liu@connect.polyu.hk (C. Liu).

https://doi.org/10.1016/j.patcog.2026.113290
Received 11 June 2025; Received in revised form 8 February 2026; Accepted 9 February 2026
Available online 11 February 2026

pairwise edges (e.g., (A, B), (A, C), ...), however this does not explicitely represent the fact that the four patients share a particular combination of biomarkers. Therefore, the use of many-to-many connections allows the model to learn representations of specific, high-order clinical subtypes rather than just abstract pairwise similarities, capturing the complex and heterogeneous nature of the disease.

Furthermore, the hypergraph structure is inherently resilient to missing data. Let us assume Patient D is missing their PET scan. In a standard graph, the pairwise similarity between D and other nodes would be calculated from incomplete features. In the hypergraph, Patient D can still be included in the hyperedge $e = (A, B, C, D)$ if their available modalities (MRI and CSF) are sufficiently similar to the group pattern. This ability to integrate information from multiple modalities simultaneously makes hypergraphs more resilient to missing data compared to traditional graphs.

A promising solution to the issue of data availability (e.g., due to patient dropouts) is instead prompt learning [10,11], which has shown great success in natural language processing (NLP) [10]. The strategy of prompt learning is “pre-training, prompting, and finetuning". This is related to transfer learning [12], which has been widely used in disease prediction. Prompt tuning involves designing or learning task-specific prompts that guide a pre-trained model to adapt to new tasks without modifying its core parameters. Unlike fine-tuning, where the model weights are updated to learn task-specific representations, prompt tuning keeps the model parameters frozen and focuses instead on learning a small number of task-specific parameters, often referred to as “prompts," which are appended to the model input. This distinction makes prompt tuning significantly more parameter-efficient compared to fine-tuning. Fine-tuning typically requires retraining large portions of the model and storing a separate set of parameters for each task, whereas prompt tuning only adjusts and saves the lightweight prompt parameters. In our case, we only need to maintain two separate set of prompts with one model rather than two models. Note also that, when the downstream task labels are noisy or there is a strong class imbalance, full fine-tuning can end up destroying the representations learned during pre-training. Prompt tuning instead serves as a form of regularization preventing the model from forgetting these biologically meaningful features. Finally, prompt tuning is designed to work effectively with limited labeled data [11], making it particularly well-suited for scenarios where the data of each downstream task is limited or the sample distribution is imbalanced.

However, adapting prompt learning to hypergraph based disease prediction is a non-trivial problem. NLP prompts operate in a space with a defined token order and clear insertion points (e.g., prepending), but hypergraphs lack these properties. First of all, a hypergraph has no sequential order, making it unclear where the prompt should be inserted. Furthermore, while NLP prompts interact via self-attention, it is unclear how prompt tokens should participate in the message-passing paradigm. Finally, unlike the simple 1D order of NLP prompts, the internal structure a set of hypergraph prompts is an open problem. Therefore, one needs to reformulate language prompts as hypergraph prompts.

To tackle the above issues, we introduce Prompted HyperGraph Neural Network (PHGNN), a framework for AD diagnosis and MCI conversion prognosis which uses MRI, PET, and non-imaging clinical data for hypergraph pre-training followed by prompt tuning. This in turn allows us to simultaneously tap into multiple heterogenous modalities in an effective way while handling missing patient data and being able to harness the predictive power of models pre-trained on large datasets. Specifically, we make the following contributions:

- We propose to use HyperGraphMAE, a simple masked autoencoder strategy to pre-train our HGNN. We then extend the concept of prompt learning, originally introduced in the context of NLP, to HGNNs, enabling us to efficiently train our model with limited data;
- We perform an extensive ablation study to evalue the impact of our architectural choices and the sensitivy of the model to its hyperparameters. We then conduct thorough experiments comparing PHGNN against other multimodal approaches and tuning strategies, demonstrating its ability to outperform SOTA alternatives in terms of both classification performance and the number of tunable parameters;
- We evaluate the interpretability of our model using SHAP analysis and by visualizing the feature embeddings. Finally, we extend the evaluation to external datasets, showing the model robustness to domain shifts and its effectiveness in handling severe data missingness.

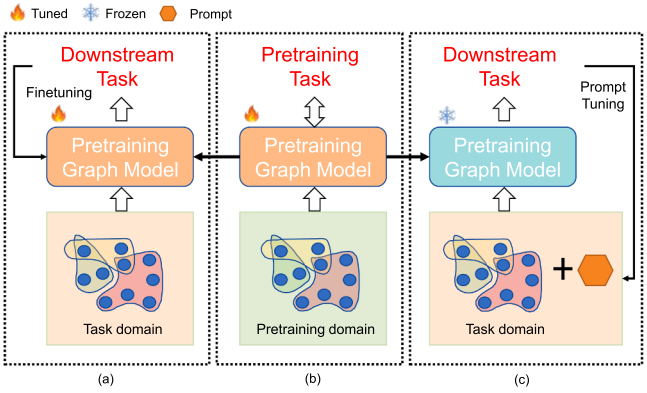


Fig. 1. Given a pre-trained model (b), in fine-tuning the model is directly tuned to solve the downstream task (a) while in prompt tuning the model is kept fixed while lightweight learnable prompts are added to the input data (c).

2. Related work

2.1. Graph neural networks

GNNs are models designed to process graph data, where nodes represent entities and edges represent relationships [13]. By iteratively aggregating information from neighboring nodes through a process known as message passing, GNNs learn rich node representations that capture the graph local structure. Various GNN architectures have been proposed to enhance the learning capabilities of these models. Graph Convolutional Networks (GCNs) [13] apply convolutional operations to graphs, generalizing the concept of convolution from grid data to graph data. Graph Attention Networks (GATs) [14] introduce attention mechanisms to weight the importance of neighboring nodes dynamically. GraphSAGE [15] aggregates neighborhood information using mean, LSTM, or pooling methods, making it scalable to large graphs. HGNNs extend traditional GNNs by modeling higher-order relationships [16], where hyperedges can connect multiple nodes simultaneously, enabling richer and more complex representations in multimodal data [17]. To enhance the learning ability of GNN, HyGNN [18] makes the aggregation function learnable by introducing an attention mechanism that weights the messages during aggregation.

2.2. GNNs for AD diagnosis

Recent advancements in AD diagnosis have successfully leveraged the ability of GNNs to model complex relationships within brain networks [19–21]. Traditional methods, such as MRI-based image analysis and machine learning models, have shown promise but often fail to capture the intricate connectivity patterns in brain regions that are critical for AD diagnosis. Early approaches to AD diagnosis using GNNs focused on modeling brain connectivity networks, where nodes represent brain regions, and edges represent functional or structural connections between them. Li et al. [19] applied GCNs to neuroimaging data, using the brain structural connectivity graph to classify AD and healthy controls. This model learns to identify distinctive patterns of degeneration based on the spatial relationships between brain regions. Recent works

have extended these ideas by incorporating multimodal data, such as MRI scans, PET images, and genetic information, into GNN frameworks. Zhang et al. [20] proposed a multi-view GNN model that combines functional and structural neuroimaging data for more accurate AD classification. These multimodal approaches enhance the GNN ability to capture both local and global brain network patterns, improving diagnostic performance. Other studies have focused on temporal dynamics, where GNNs are used to model disease progression over time. For instance, Shen et al. [21] utilized dynamic GNNs to track changes in brain connectivity over time, enabling early detection and more precise prediction of AD progression.

2.3. Prompt learning

Prompt learning is an emerging paradigm in machine learning, particularly popularized in NLP by models like GPT [22]. Instead of fine-tuning an entire model for specific tasks, prompt learning guides the model to generate task-specific outputs by crafting input prompts [23]. This approach has been adapted to various tasks, including text classification, question answering, and generation, and has shown that large pre-trained models can perform well with minimal task-specific modifications [24].

Recent research has extended prompt learning beyond NLP to other domains, including computer vision [25] and graph-based tasks [26]. Here prompt learning often involves designing prompts that condition the model to focus on relevant features or substructures within the data, enabling the model to generalize across different tasks with limited supervision. In the context of GNNs, prompt learning has been applied to guide the model in learning specific graph patterns or node attributes with fewer labeled examples, enhancing efficiency and performance. As shown in Fig. 1, prompt tuning focuses on developing lightweight learnable prompts while keeping the pre-trained model unchanged. This makes prompt tuning very effective for few-shot downstream tasks [27]. In the graph domain, the design of prompts usually takes two main forms: (1) the prompt can either be treated as a set of additional learnable parameters that are added to the original node features [26], or (2) it can be viewed as the feature of a super node connected to all the nodes of the input graph. In the latter, optimizing the feature of this super node can be treated as adding a global receptive field learning a representation for the whole graph [28].

3. Prompted hypergraph neural network

PHGNN is a semi-supervised prompted hypergraph learning framework composed of 3 key components: 1) modality-aware hypergraph representation, 2) self-supervised hypergraph pre-training, and 3) hypergraph prompt learning. Fig. 2 shows an overview of the structure of our framework.

3.1. Hypergraph construction

Unlike graphs, where edges connect two nodes, the hyperedges of a hypergraph connect multiple nodes, enabling the representation of higher-order relationships between node subsets with shared features which can model complex interactions among the input modalities. In particular, an undirected node-attributed hypergraph is defined as $G = (V, E, H)$, with node set $V = \{v_1, v_2 \dots, v_{|V|}\}$, hyperedge set $E = \{e_1, e_2 \dots, e_{|E|}\}$, and incidence matrix $H \in \{0,1\}^{|V|\times|E|}$. Each node $v_i \in V$ corresponds to a subject (i.e., a patient) with node feature $x_i \in \mathbb{R}^d$, where d is the dimension of the feature, and the number of nodes $|V|$ equals the number of patients. Similarly, we denote node feature matrix as $X \in \mathbb{R}^{|V|\times d}$. For each node v_i, we create a hyperedge $e_i \in E$ by connecting v_i to its k-nearest neighbors based on the Euclidean distance between node features. The incidence matrix H is an alternative way to represent E, where $H(v_i, e_i) = 1$ if $v_i \in e_i$, or $H(v_i, e_i) = 0$ otherwise.

Recall that the input data is composed m modalities $I = \{I_1, I_2, \dots, I_m\}$. For each modality we feed the input data to the corresponding pre-trained feature extraction backbone and we obtain m sets of node features $X = \{X_1, X_2, \dots, X_m\}$, where $X_i \in \mathbb{R}^{|V_i|\times d_i}$ and $|V_i| = |V_j|$, $d_i = d_j$ for all $1 \le i, j \le m$. For each modality $X_i = \{x_i^1, x_i^2, \dots, x_i^{|V_i|}\}$, we build a hypergraph G_i by constructing the corresponding set of hyperedges E_i using the k-nearest neighbour (k-NN) method, as discussed above. For each modality, this results in $|V_i|$ hyperedges, each linking $k+1$ nodes. Consequently, we obtain an incidence matrix $H_i \in \mathbb{R}^{|V|\times|V|}$, where $|V_i| \times (k+1)$ entries are set to 1, and all other entries are 0. The resulting hypergraphs from each modality are finally concatenated to obtain the fused hypergraph $G = (V, E, H)$ using the *coequal fusion mechanism* introduced in [8], which treats each modality equally and integrates both the *features* and the *structural relationships* from all modalities.

Specifically, the fused node feature matrix $X \in \mathbb{R}^{|V|\times d}$ for the hypergraph is created by concatenating feature matrices from each of the m modalities along the feature dimension, where $|V| = |V_i|$ for all $1 \le i \le m$ and $d = \sum_{i=1}^{m} d_i$ (the sum of feature dimensions across all m modalities), i.e., $X = \text{concat}_{\text{axis}=1}(X_1, X_2, \dots, X_m)$. The fused hyperedge set E is then created by taking the union of all the modality-specific hyperedge sets, i.e., $E = E_1 \cup E_2 \cup \dots \cup E_m = \bigcup_{i=1}^{m} E_i$. The fused incidence matrix $H \in \mathbb{R}^{|V|\times(m|V|)}$ is constructed by concatenating the individual incidence matrices $\{H_1, H_2, \dots, H_m\}$ along the column dimension (hyperedge dimension), i.e., $H = \left[H_1 \parallel H_2 \parallel \dots \parallel H_m\right]$, where $\parallel$ denotes the matrix concatenation operation. Consequently, the columns of H correspond to the complete set of hyperedges in E.

3.2. Hypergraph neural network

In this paper, we follow the work of [29], where the authors use hypergraph convolutional layers to capture hypergraph features. In the hypergraph convolutional layer, message passing occurs either from a vertex to a hyperedge or from a hyperedge to a vertex.

First we perform a *vertex-to-hyperedge convolution*, where the information of all vertices in a hyperedge is aggregated. This step can be understood as integrating the features of a group of subjects with high similarity onto a single hyperedge to capture high-order correlations. Let $Z^{(l)}$ denote the aggregated hyperedge information derived from the $(l-1)$th layer vertex features $X^{(l-1)}$ and D_v be the diagonal matrix of vertex degrees (the number of hyperedges each vertex belongs to), then $Z^{(l)} = H^T D_v^{-\frac{1}{2}} X^{(l-1)}$.

A *hyperedge-to-vertex convolution* is then performed, where the vertex learns from the hyperedges it is connected to. This updates the vertex own feature, which can be interpreted as the subject refining its representation based on the integrated information of the groups (hyperedges) it belongs to, i.e., $\hat{X}^{(l)} = D_v^{-\frac{1}{2}} H D_e^{-1} Z^{(l)}$, where D_e is the diagonal matrix hyperedge degrees (the number of nodes contained in each hyperedge).

These two stages update the vertex features X through message propagation, and are combined into to a two-stage hypergraph convolution operation with learnable filters, $X^{(l)} = \sigma(D_v^{-\frac{1}{2}} H D_e^{-1} H^T D_v^{-\frac{1}{2}} X^{(l-1)}\Theta^{(l)})$, where $\Theta^{(l)}$ 'represents the trainable parameters of the layer and σ is a non-linear activation function. The HGNN performs hypergraph convolutions followed by an output layer producing the subject classification $Y \in \mathbb{R}^{|V|\times 2}$, i.e., $Y = \text{Softmax}\big(\mathcal{A}\,\text{ReLU}\big(\mathcal{A}X^{(0)}\Theta^{(0)}\big)\Theta^{(1)}\big)$, where $\mathcal{A} = D_v^{-\frac{1}{2}} H D_e^{-1} H^T D_v^{-\frac{1}{2}}$.

3.3. HyperGraphMAE pre-training

Inspired by GraphMAE [30], we propose HyperGraphMAE, a generative self-supervised learning framework that we use to pre-train the HGNN at the core of our pipeline. Given $G = (V, E, H)$, let h_E be a hypergraph encoder, h_D a hypergraph decoder, and $O \in \mathbb{R}^{|V|\times d_z}$ the

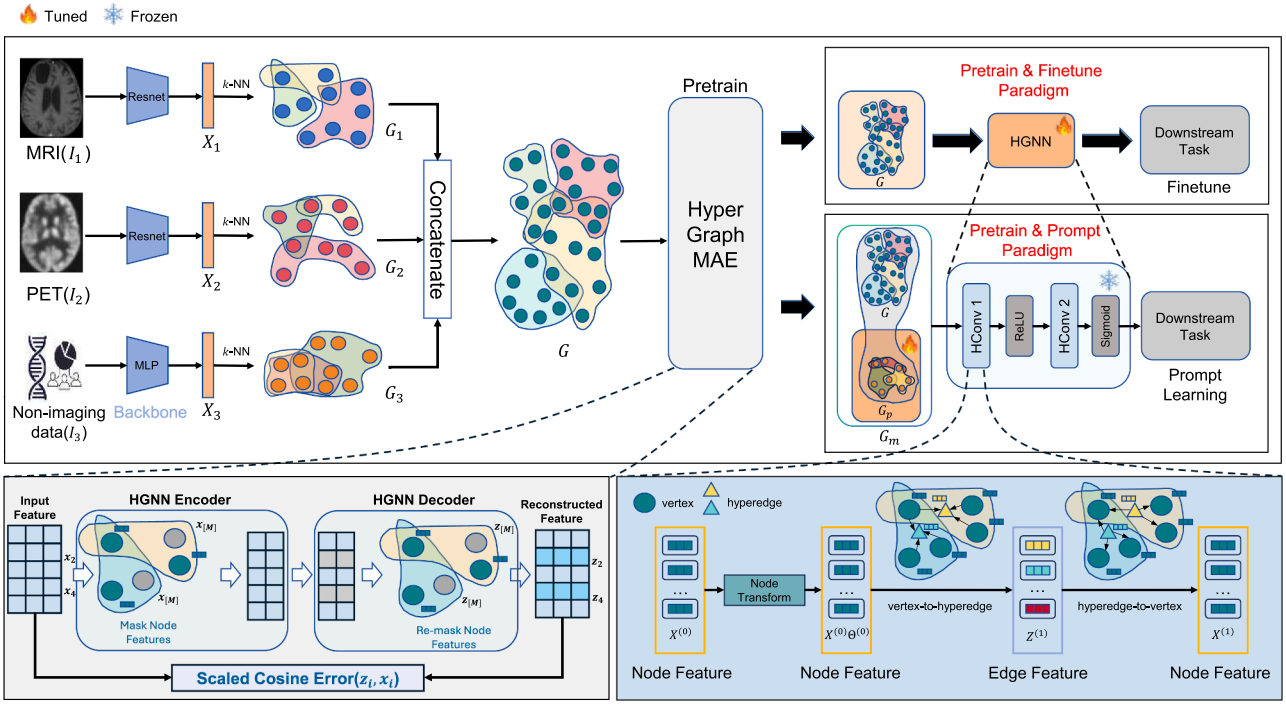


Fig. 2. The proposed PHGNN framework.

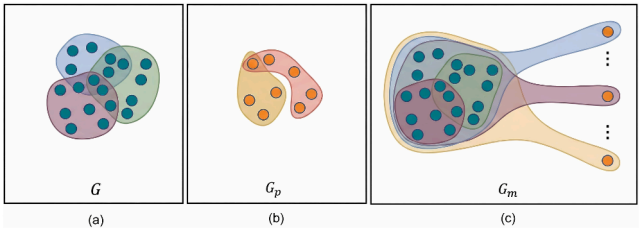


Fig. 3. (a) A small example input hypergraph G; (b) a small prompt sub-hypergraph G_p with a few tokens; and (c) the resulting manipulated hypergraph G_m, visually highlighting the new hyperedges that connect each prompt token to all the nodes in G.

corresponding hidden state, i.e.,

$$O = h_E(G, X), \quad G' = h_D(G, O), \tag{1}$$

where G' is the reconstructed hypergraph, and both h_E and h_D are expressive HGNNs capable of leveraging information from the node neighbourhood.

HyperGraphMAE is trained to reconstruct masked node features of the hypergraph G. To this end, during the encoding phase a subset of nodes $V_m \subset V$ of G is randomly selected and their features are masked with a learnable token $x_{[M]}$. After the encoding, the embeddings of the masked nodes are re-masked with another token $o_{[M]}$ before being passed to h_D. This encourages the decoder to reconstruct node features using information from neighboring nodes. The goal is then to reconstruct the masked node features $\hat{X}$ given the partially observed node features X and the adjacency matrix H of G. We use the Scaled Cosine Error (SCE) as the reconstruction loss, which has the advantage of reducing the sensitivity and selectivity issues of the mean squared error, i.e.,

$$L_{\text{SCE}} = \frac{1}{|V_m|} \sum_{v_i \in V_m} \left(1 - \frac{x_i^\top x_i'}{\|x_i\| \|x_i'\|}\right)^\gamma,$$

where x_i and x_i' are the original and reconstructed node features, respectively, and $\gamma \geq 1$ is a scaling factor that improves selectivity by down-weighting the contribution of easy samples during training.

3.4. Hypergraph prompt

In this paper, we develop a method to seamlessly transfer the idea of prompting from NLP to hypergraph learning. In NLP, a prompt consists of a linear sequence of prompt tokens, resembling a sub-sentence or phrase. Typically, the prompt is appended to either the beginning or the end of the input sentences. As such, adapting the concept of prompts to hypergraphs requires defining three key elements: (1) prompt token, (2) prompt structure, and (3) pattern insertion. Fig. 3 illustrates an example of our prompt design.

1. Prompt tokens. Given a hypergraph $G = (V, E, H)$, we introduce the prompt sub-hypergraph $G_p = (P, E_p, H_p)$, where $P = \{p_1, p_2, \ldots, p_{|P|}\}$ represents the set of $|P|$ learnable prompt tokens and E_p is the hyperedge set inside the prompt sub-hypergraph. Each token $p_i \in P$ is represented as a token vector $p_i \in \mathbb{R}^{1\times d}$, matching the dimensions of the node features in the input hypergraph. In practice, $|P|$ is much smaller than N and $|P| \ll d_z$, where d_z is the hidden layer size in the pre-trained hypergraph model.

2. Prompt structure. We model the relationship among prompt tokens using k-NN based on the Euclidean distance between the tokens. This results in an incidence matrix H_p for the prompt sub-hypergraph G_p. Note that H_p is recomputed at the start of every training epoch given the latest prompts P.

3. Pattern insertion. Let ψ represent the insertion method to add the prompt graph G_p to the input hypergraph G, resulting in the manipulated hypergraph $G_m = \psi(G, G_p)$. We leverage the fact that a hyperedge can connect multiple nodes and for each prompt token we define a hyperedge connecting it to all the nodes of G, i.e., $E_m = E \cup E_p \cup \left\{\{v_1, v_2, \ldots, v_N, p_i\}, \forall p_i \in P\right\}$, where E_m, E, and E_p are the sets of hyperedges of G_m, G, and G_p, respectively, and $\{v_1, v_2, \ldots, v_N, p_i\}$ denotes a new hyperedge connecting the node corresponding to the token p_i of G_p to the N nodes of G.

3.5. Overview of the prompt learning process

The main objective of our framework is a binary class prediction task (AD vs CN, sMCI vs pMCI). The overall prompt learning process starts with a hypergraph G (see Section 3.1) and a pre-trained HGNN model h_θ, where the pre-trained weights are from the HyperGraphMAE encoder h_E of Eq. (1) and are frozen during prompt learning. To split the data into train and validation sets, we define the training mask M_T and the validation mask M_V, which are binary masks used to control which nodes of the graph are used during different phases of the training process.

Given this setting, a sub-hypergraph G_p with a set of prompt tokens is initialized and constructed. Then G and G_p are concatenated through hyperedges to produce the hypergraph G_m, which is processed by the frozen pre-trained HGNN model h_θ and only the learnable prompt tokens P are optimized. Note that, for each epoch, G_p is reconstructed again with the updated set of tokens P. The core intuition for designing G_p as a dynamic graph is to treat prompt tokens not as independent, isolated vectors, but as a self-organizing, collaborative system. Firstly, complex downstream tasks like AD classification often involve heterogeneous sub-patterns within a single class. Different prompt tokens can specialize into distinct sub-patterns. The internal structure G_p acts as a communication mechanism, allowing these specialized tokens to share information and coordinate, thereby forming a more comprehensive prompt for the task. Secondly, in the initial phase of training, prompt tokens are randomly initialized, and their relationships are therefore random. As the training progresses, they gradually learn meaningful semantic representations. Re-calculating k-NN ensures that their interactions are always based on their most recent semantic similarity, rather than being constrained by an initial random state.

The resulting output is a probability distribution over the target classes. We use the supervised cross-entropy (CE) loss $\mathcal{L}_{sup} = \sum_{i=1}^{N} CE(\hat{y}_i, y_i)$, where $\hat{y}_i$ represents the output probability and y_i corresponds to the ground truth label for the ith node (i.e., the ith patient).

4. Experimental results

Data description. We evaluate our framework on the Alzheimer's Disease Neuroimaging Initiative (ADNI) [31]. The ADNI project gathered brain images and clinical data from thousands of participants across North America. Initially launched as a five-year study (ADNI-1), it was extended in 2009 (ADNI-2). ADNI-3 was instead officially launched in September 2016 and ran until 2023, adopting full 3T MRI imaging across all sites. We also validate our framework on two external datasets: Open Access Series of Imaging Studies-3 (OASIS-3) [32], an open-access multimodal neuroimaging dataset led by the Knight Alzheimer's Disease Research Center (Knight ADRC) at Washington University; the National Alzheimer's Coordinating Center (NACC) dataset [33], an open-access multimodal research dataset established by the National Institute on Aging. We employed T1-weighted MRI, FDG (fluorodeoxyglucose)-PET and their corresponding biological markers, as well as clinical and neuropsychological assessment data. In addition, we also incoporate Amyloid-PET data to evaluate the effectiveness of this type of data in the ADNI and NACC datasets.

The ADNI-1 MRI data includes scans from 821 individuals (196 with AD, 168 with pMCI, 230 with sMCI, and 227 CN (cognitive normal)) using 1.5T scanners, with FDG-PET and Amyloid-PET data available for 396 and 170 of them, respectively. For ADNI-2, baseline MRI data was acquired from 534 participants (156 with AD, 66 with pMCI, 112 with sMCI, and 200 CN) using 3T scanners, while FDG-PET and Amyloid-PET data are available for 487 and 468 of these subjects. For ADNI-3, baseline MRI data was acquired from 58 participants (8 with pMCI, 50 with sMCI) using 3T scanners, ensuring standardized high-resolution imaging. However, we omit AD classification on this dataset as there are only 2 AD subjects in the baseline scanning. Regarding OASIS-3, the baseline data was collected from 1280 individuals (240 with AD, 1040 with CN), where the problem of missing modality is particularly severe, with only 110 subjects having associated PET data. Since only labels for the AD classification task are available, we did not consider OASIS-3 dataset for the MCI conversion prediction task. Finally, the NACC dataset includes 3264 individuals (344 with AD, 154 with pMCI, 294 with sMCI, and 2472 CN), with 230 FDG-PET scans and 687 Amyloid-PET scans.

Implementation details. In our experiments we use three modalities: MRI, PET, and tabular data. For imaging data (MRI, FDG-PET and Amyloid-PET) in ADNI, OASIS, and NACC we adopt the same preprocessing procedure as [3] and make use of the FMRIB Software Library version 6.0.3. We first removed non-brain tissue from the raw data using the Brain Extraction Tool. We then employed the FLIRT algorithm to perform linear spatial normalization, aligning the MRI scans to the standard MNI152 T1 template. Simultaneously, PET scans were co-registered to their respective subject-specific T1-weighted images and subsequently transformed into the common MNI coordinate space. To eliminate irrelevant background information, the volumes were cropped to dimensions of $152 \times 188 \times 152$ by discarding peripheral zero-valued voxels. Finally, to decrease the computational overhead for the proposed model, the inputs were downsampled to a resolution of $76 \times 94 \times 76$. For the tabular data, we use the same set of variables and pre-processing principles as [2], where we normalize each numerical value using min-max scaling. We augment the feature set with binary indicators to encode missingness for all tabular attributes, excluding age, gender, and education, which are fully observed. We use a ResNet-50 to extract features from MRI and PET images and a multi-layer perceptron to extract features from the tabular data. Given these features, the hypergraph representing the set of patients is constructed by setting the k-NN parameter to 30 (see Section 3.1). We optimize the model using AdamW with a weight decay of 1e-4 and a learning rate of 3e-4.

For ADNI-1, we further split the data using a 5-fold cross-validation strategy. In addition, pre-training and finetuning are performed over the same training subjects. We experimentally tuned the value of $|P|$ as discussed in the ablation study. Based on these results, we fixed the number of prompt tokens to $|P| = 16$. Furthermore, the internal structure of the prompt sub-hypergraph (G_p) was constructed using a k-NN approach with $k_{prompt} = |P|/4 = 4$. For HyperGraphMAE, we use a uniform random sampling strategy without replacement, with 75% of the nodes being masked. We set γ in L_{SCE} be 2. Our code can be accessed at anonymous.4open.science/r/PHGNN-B3B3/. All experiments were performed on a server powered by an AMD EPYC 7302 16-Core Processor and a NVIDIA GeForce RTX 4090 GPU.

Evaluation protocol. In line with other works in this area [3,34], we train our model on the ADNI-1 dataset and evaluated its performance using the ADNI-2 dataset. We also evaluate the ability of our model to generalize to external datasets and cope with potential domain shift issues. To this end, we evaluate our model trained on ADNI-1 on two external datasets, ADNI-3, OASIS-3, and NACC. Specifically, we evaluate our model on multimodal classification tasks, specifically AD vs CN and pMCI vs sMCI, against SOTA alternatives as well as other graph parameter-efficient fine-tuning (PEFT) methods. To assess the performance of the proposed method, we calculate four metrics with their average and standard error over five folds: balanced accuracy (BACC), sensitivity (SEN), specificity (SPE), and area under the curve (AUC).

4.1. Comparison with existing methods

We compare the classification performance of PHGNN with that of three SOTA multimodal classification models (PT-GCN [3], SPDN [34], and Modality-Flexible Framework (MFF) [4]), two hypergraph based models based (HGNN [17] and HGNN + [29]), and a GNN based method [35]. The hypergraph based models mirror our pipeline without resorting to prompt tuning. HGNN + [29] is an improved version of HGNN obtained by introducing hyperedge groups with adaptive fusion for better handling of diverse information and by replacing spectral convolution

with a flexible, two-stage spatial-domain convolution that is extendable to directed hypergraphs. The GNN, HGNN, and HGNN+ models are also pre-trained using the HyperGraphMAE method for fair comparison. We run MFF using publicly available code, while we use the reported results for PT-GCN and SPDN as they are based on exactly the same ADNI subjects and data split setting we use.

Table 1 shows that PHGNN outperforms the other models, achieving the highest BACC and AUC in both tasks, showing its ability to distinguish between AD and CN effectively. We also observe a clear improvement of PHGNN over HGNN+, indicating the effectiveness of our prompting strategy. In the more challenging task of pMCI vs sMCI, PHGNN again shows superior performance, outperforming MFF across all metrics. Note that we use HGNN and not HGNN+ in PHGNN as the former has higher AUC in this task. This indicates that PHGNN effectively captures subtle patterns in the data, making it well-suited for complex clinical prediction tasks. Compared to previous methods like HGNN and HGNN+, PHGNN introduces more effective hypergraph construction and optimization, which contribute to its superior performance, especially in handling nuanced clinical classification tasks.

4.2. Comparison with other PEFT methods

To evaluate the effectiveness of our prompting strategy, we compare it with two other prompt learning methods that can be directly used with hypergraphs, GPF and GPF-Plus [26], and two adapter methods LoRA [36] and AdapterGNN [37], as well as the Pre-train & Finetune approach. GPF and GPF-plus primarily incorporate soft prompts into all node features of the input graph. The results are shown in Table 2. In the AD vs CN task, PHGNN achieves the best performance across all metrics. Compared to the traditional Pre-train & Finetune approach, PHGNN improves both classification accuracy and the balance between sensitivity and specificity. In general, our experiments show that PHGNN outperforms all competing PEFT methods while offering a significant advantage in terms of parameter efficiency (see Section 4.7).

4.3. Comparison with other pre-training methods

To assess different self-supervised pre-training strategies for PHGNN, we compare four representative methods-SimGRACE [38], DGI [39], HyperGCL [40], and our HyperGraphMAE-on two classification benchmarks (AD vs CN and pMCI vs sMCI), as summarized in Table 3.

SimGRACE generates correlated graph views by injecting stochastic noise into the encoder and minimizes an InfoNCE contrastive loss [38], while DGI [39] maximizes the mutual information between local node embeddings and a global summary vector. HyperGCL [40] is instead a self-supervised framework for hypergraph representation learning that applies contrastive learning to hypergraphs by constructing augmented views via both fabricated and generative schemes. Compared to these pre-training strategies, HyperGraphMAE achieves the best overall performance on both AD vs CN and on pMCI vs sMCI demonstrating that hyperedge-level reconstruction provides the most discriminative pre-training representations for PHGNN across both AD diagnosis and MCI progression prediction. Moreover, the results show that combining PHGNN with different pre-training methods consistently outperforms other PEFT methods, demonstrating the robustness of our methods.

4.4. Experiments on external datasets

To further evaluate the generalization capability and robustness of PHGNN, we perform additional experiments on two external datasets: ADNI-3 for sMCI vs pMCI classification and OASIS-3 for AD vs NC classification. We compare PHGNN against previous baselines, as presented in Table 4. For a fair comparison, all models are trained on ADNI-1.

On the ADNI-3 (sMCI vs pMCI) task, PHGNN again demonstrates superior performance, achieving the highest BACC and AUC and generally displaying a robust ability to correctly identify MCI subjects in this newer cohort. Similarly, in the AD vs NC classification task on the OASIS-3 dataset, PHGNN confirms its strong generalization ability. It achieves the highest BACC and AUC, highlighting its ability to cope effectively with different demographics and acquisition protocols. The strong performance on OASIS-3 is particularly noteworthy as this dataset suffers from a high proportion of missing PET scans ($\sim 90\%$).

To further evaluate the cross-domain performance of PHGNN, we consider the NACC dataset and introduce a new clinical task: differentiating pMCI from NC. The results are summarized in Table 5. Distinguishing pMCI subjects from NC is critical for early intervention. On this task, PHGNN consistently outperforms competing baselines on both the ADNI-2 and external NACC cohorts. While SOTA methods like MFF and AdaGNN achieve competitive results, our framework demonstrates a clear performance edge, confirming that its discriminative power transfers effectively to the task of identifying early-stage pathological deviations from NC. We finally evaluated pMCI vs sMCI on the external NACC dataset by incorporating FDG-PET or Amyloid-PET. Our PHGNN shows consistent improvement in both scenarios.

Overall, the consistent high performance of PHGNN across both the ADNI-3, OASIS-3, and NACC external datasets strongly validates our approach. The results show that PHGNN not only excels on the primary dataset but also demonstrates significant generalization and robustness, effectively capturing the complex data patterns required across different patient cohorts.

4.5. Ablation study

We then perform an ablation study on the sMCI vs pMCI task in order to evaluate the effect of: 1) varying the number of prompt tokens $|P|$; 2) replacing the HGNN with a GNN denoted as Prompt GNN; 3) replacing our *Prompt as graph* strategy, where a graph prompt has multiple prompt tokens and a non-trivial structure, with *Prompt as token*, where we treat prompt tokens as independent prompts without considering the inner structure (denoted as PHGNN w/o S); 4) varying the k-NN parameter when constructing the hypergraph; 5) training on different ratios of data; 6) sparser and more localized prompting strategy.

Table 6 shows the results of the first four ablation studies. Firstly, we observe that the number of tokens $|P|$ can have a significant influence on the result of the prompt learning, which is in accordance with what is highlighted in [41]. Therefore, to get optimal results it is important to select the optimal number of tokens $|P|$. In future work, one possibility would be to leverage meta-learning to automatically select the best value of $|P|$. Secondly, although the prompt learning strategy leads to an improvement over the original GNN performance (see Table 1), this is still inferior when compared to PHGNN. Moreover, the results clearly demonstrate the importance of the inner structure of prompt graphs, as opposed to simply treating the tokens as independent prompts.

Moreover, Table 6 (Right) shows the impact of the choice of k in the k-nearest-neighbor graph used to build the hypergraph for PHGNN. As k increases from 10 to 30, the AUC steadily rises , demonstrating that incorporating a moderate number of neighbors captures the most informative higher-order relationships. When k grows beyond 30, performance slightly decreases , indicating that overly dense connectivity introduces noise and degrades representation quality. Based on these results, we adopt $k = 30$ for all subsequent experiments.

We also evaluate the impact of using different ratios of training data on the performance of HGNN and PHGNN. Fig. 4 shows that by using just 20% of the training data PHGNN can still achieve a performance comparable to that of HGNN trained on the full dataset, thus validating the effectiveness of our prompt tuning strategy under limited data scenarios.

Lastly, we test the use of sparser connections between prompts and nodes. Specifically, for each prompt, we connect a subset of the nodes based on feature similarity ranging from 20% to 80% (stride = 20%), rather than connecting it to all the nodes (100%). As shown in Fig. 4, connecting all the nodes for each prompt results in the best performance.

Table 1
Classification performance on AD vs CN and pMCI vs sMCI. For each metric we show the average (± std deviation) over 5 folds (best model highlighted in green).

	AD vs CN				pMCI vs sMCI			
	BACC	SEN	SPE	AUC	BACC	SEN	SPE	AUC
GNNs [35]	86.8±0.3	89.7±0.2	83.7±0.4	91.5±0.4	70.9±0.7	52.7±0.7	89.1±0.6	75.9±0.5
HGNN [17]	89.7±1.3	88.7±1.0	90.7±1.2	94.1±0.8	74.7±0.7	72.3±0.9	77.1±0.4	77.7±0.8
HGNN+ [29]	89.4±0.9	91.6±1.1	87.1±0.9	94.2±1.2	74.6±1.3	77.5±1.2	71.8±1.3	77.3±1.6
PT-DCN [3]	92.7	91.7	93.5	96.4	75.3	70.8	78.4	77.8
SPDN [34]	92.9	91.9	93.6	96.6	76.2	62.8	80.6	77.3
MFF [4]	91.2±0.5	91.5±1.5	90.9±0.9	95.5±0.4	77.6±1.1	74.9±1.6	80.2±0.6	80.6±0.5
PHGNN	93.2±0.6	90.6±1.9	95.6±0.6	97.2±1.2	79.6±0.6	75.3±0.8	83.8±1.5	82.7±1.1

Table 2
Prompt learning strategies on AD vs CN and pMCI vs sMCI. For each metric we show the average (± std deviation) over 5 folds (best model highlighted in green).

	AD vs CN				pMCI vs sMCI			
	BACC	SEN	SPE	AUC	BACC	SEN	SPE	AUC
Finetune	89.7±1.3	88.7±1.0	90.7±1.2	94.1±0.8	74.7±0.7	72.3±0.9	77.1±0.4	77.7±0.8
GPF [26]	90.3±1.9	87.4±1.1	93.0±1.5	95.3±1.4	75.7±0.9	75.3±1.1	76.0±0.9	80.2±0.8
GPF-Plus [26]	89.0±1.4	89.7±1.2	88.4±1.6	93.3±1.5	71.1±0.5	47.6±0.4	94.5±0.8	79.0±1.8
LoRA [36]	89.0±0.1	86.1±0.3	91.8±0.2	93.0±0.1	76.2±0.5	70.7±0.7	81.7±0.5	79.5±1.0
AdaGNN[37]	90.3±4.0	86.1±1.3	94.5±0.6	94.5±0.1	77.6±0.6	69.2±1.4	86.0±0.5	80.5±0.2
PHGNN	93.2±0.6	90.6±1.9	95.6±0.6	97.2±1.2	79.6±0.6	75.3±0.8	83.8±1.5	82.7±1.1

Table 3
Pre-training strategies on AD vs CN and pMCI vs sMCI. For each metric we show the average (± std deviation) over 5 folds (best model highlighted in green).

	AD vs CN				pMCI vs sMCI			
	BACC	SEN	SPE	AUC	BACC	SEN	SPE	AUC
SimGRACE [38]	91.5±0.6	91.3±0.8	91.7±0.6	95.8±0.5	76.3±0.7	63.9±0.6	88.6±0.8	81.5±0.4
DGI [39]	92.4±0.8	91.5±1.1	93.4±0.9	96.1±0.9	77.0±0.5	64.8±1.3	89.1±0.9	80.8±0.5
HyperGCL [40]	92.2±0.4	92.9±0.6	91.5±0.7	96.2±0.3	77.7±0.7	65.2±0.9	90.2±0.5	81.6±0.8
HyperGraphMAE	93.2±0.6	90.6±1.9	95.6±0.6	97.2±1.2	79.6±0.6	75.3±0.8	83.8±1.5	82.7±1.1

Table 4
Classification performance over 5 folds (average ± std deviation, best results highlighted in green) on ADNI-3 (sMCI vs pMCI) and OASIS-3 (AD vs NC).

Method	ADNI-3 (sMCI vs pMCI)				OASIS-3 (AD vs NC)			
	BACC	SEN	SPE	AUC	BACC	SEN	SPE	AUC
GNNs	66.0±0.3	67.0±1.4	75.9±1.5	68.9±0.3	65.3±0.6	67.7±0.6	63.0±0.5	73.1±0.7
MFF	74.0±0.1	64.6±0.6	83.4±0.5	76.8±0.2	73.9±0.6	71.2±1.4	76.6±1.1	80.1±0.8
HGNN	72.1±0.2	60.0±0.3	84.3±0.4	74.9±0.2	72.6±0.2	71.4±2.4	73.7±2.6	78.9±0.2
GPF	72.4±0.2	64.6±0.6	80.2±0.3	74.2±0.2	73.2±1.6	66.0±5.9	80.4±6.1	79.4±1.2
GPF-Plus	72.0±2.1	64.6±0.6	79.5±0.2	74.4±0.1	71.8±2.1	75.7±3.8	68.0±3.5	78.0±1.9
LoRA	73.9±0.1	63.0±0.3	84.9±0.4	76.2±0.1	73.3±0.2	72.8±4.4	73.7±4.4	78.8±0.6
AdaGNN	73.2±0.1	61.5±0.2	84.8±0.2	74.7±0.1	74.5±0.1	71.3±0.3	80.0±0.4	80.5±0.1
PHGNN	76.4±0.2	61.5±0.1	91.3±0.2	79.3±0.1	76.3±0.3	77.9±3.1	74.6±3.1	81.8±0.2

Table 5
Classification performance over 5 folds (average ± std deviation, best results highlighted in green) on ADNI-2 and NACC datasets.

Methods	pMCI vs NC				pMCI vs sMCI			
	ADNI-2		NACC		NACC (FDG)		NACC (Amyloid)	
	BACC	AUC	BACC	AUC	BACC	AUC	BACC	AUC
MFF	79.4±1.1	83.6±2.0	77.9±0.8	83.4±0.6	67.2±2.5	67.9±3.2	71.4±2.6	72.3±4.1
HGNN	78.8±0.9	82.3±1.9	77.4±1.4	83.2±0.8	67.8±0.4	67.9±0.5	69.6±0.6	71.2±0.4
GPF-Plus	76.5±1.3	82.9±1.6	75.0±0.4	80.0±0.1	67.2±1.3	70.1±1.5	69.9±0.4	71.3±0.3
LoRA	76.8±1.1	82.0±1.2	76.7±1.0	82.0±0.3	66.3±1.0	71.2±1.4	70.4±1.4	72.2±0.8
AdaGNN	78.2±1.0	83.1±1.1	77.1±1.1	83.9±1.4	66.1±2.1	67.4±2.6	69.8±1.9	73.1±2.4
PHGNN	84.8±0.5	87.7±1.2	81.0±0.5	86.9±0.7	71.8±1.2	75.6±0.9	75.1±1.5	80.1±1.4

Table 6
(**Left**) AUC scores for various settings on sMCI vs pMCI (best model in green). (**Right**) AUC scores for different k-NN on sMCI vs pMCI (best model in green).

	AUC			
$\|P\|$	8	16	32	64
Prompt GNN	–	77.4	–	–
PHGNN w/o S	–	79.8	–	–
PHGNN	76.8	82.7	81.4	72.8

	AUC				
k	10	20	30	40	50
PHGNN	81.3	81.7	82.7	82.2	82.0

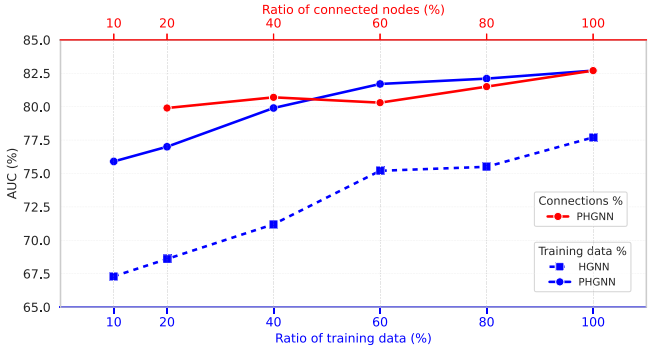


Fig. 4. (Blue) The result of using different ratio of training data for training (dashed) HGNN, (solid) PHGNN. (Red) The result of connecting different ratio of nodes for each prompt based on feature similarity in PHGNN.

We assume that this is because our prompts require a global perspective, i.e., they must be able to integrate information from the entire graph (all subjects) to most effectively learn how to align different modalities and handle missing data, thereby achieving optimal performance. A sparse or local prompt connection restricts the model ability to acquire contextual information, leading to a decline in performance.

4.6. Contribution of different modalities

In Table 7, we evaluate the contributions of different data modalities in PHGNN. The first three rows represent diagnostic results for MCI prediction using each modality independently. FDG-PET scans outperform MRI scans, which aligns with prior research [42] suggesting that hypometabolism in PET scans appears earlier in the disease than atrophy in MRI. Non-imaging clinical data perform worse, with the lowest BACC and AUC across all modalities.

The next three rows show the results of modality fusion. Adding a modality boosts performance, demonstrating the importance of multimodal fusion and PHGNN's ability to handle it. Combining all three modalities yields the best diagnostic performance, suggesting that features from different modalities provide complementary insights, improving classification accuracy.

Furthermore, we incorporate Amyloid-PET in the last two rows of Table 7. While the BACC remains relatively stable, we observe a notable improvement in terms of AUC, indicating enhanced overall discriminative power. This effectiveness is further validated on the external NACC dataset (Table 5), where PHGNN leveraging Amyloid-PET demonstrates superior robustness compared to both baseline methods and the FDG-PET configuration.

4.7. Parameter efficiency analysis

As a final experiment (Table 8), we compare the number of tunable parameters of the proposed PHGNN model and other tuning strategies in the sMCI vs pMCI classification task: traditional fine-tuning (~ 0.5M), GPF [26] (~ 5K), GPF-plus [26] (~ 0.21M), LoRA [36] (~ 0.1M), AdapterGNN [37] (~ 0.1M), and PHGNN (~ 0.03M). The results show that PHGNN offers significant advantages in terms of parameter efficiency in addition to the performance improvement already illustrated in Table 2. Specifically, our approach only uses 6% tunable parameters compared to fine-tuning. Note also that our approach outperforms both GPF and GPF-plus by a large margin while only introducing a very small overhead, as only 5% more tunable parameters (wrt fine-tuning) are used compared with GPF.

Table 7
The results of PHGNN using different combinations of multimodal data on pMCI vs sMCI task.

MRI	FDG-PET	Amyloid-PET	Tabular	BACC	SEN	SPE	AUC
✓				72.9	66.3	79.5	77.1
	✓			74.0	67.2	80.8	77.4
			✓	72.4	65.1	79.6	76.6
✓	✓			76.0	67.2	83.9	79.6
✓			✓	73.8	69.2	78.5	78.9
	✓		✓	77.4	72.0	82.8	80.4
✓	✓		✓	79.6	75.3	83.8	82.7
✓		✓	✓	79.3	82.2	76.4	84.2
✓	✓	✓	✓	79.7	73.6	86.0	84.8

Furthermore, we evaluated the training and inference latency of multiple PEFT methods over 1000 runs. PHGNN imposes virtually no extra overhead, it achieves modest speed gains over full fine-tuning during both training and inference. When compared to other PEFT approaches, especially LoRA and AdapterGNN, PHGNN delivers substantial acceleration while simultaneously attaining the highest accuracy. This efficiency advantage is further emphasized by the analysis of memory consumption and computational load (GFLOPs). As shown in the final three columns, PHGNN is equally efficient in terms of memory usage. Its required training memory is lower than full fine-tuning and it maintains a minimal inference memory that is comparable to the baseline . In contrast, AdapterGNN and GPF-plus introduce more memory overhead. In terms of computational load, PHGNN introduces only a negligible increase in GFLOPs compared to full fine-tuning In summary, PHGNN not only excels in accuracy and parameter efficiency but also achieves an efficient overhead.

4.8. Model interpretability

Finally, we evaluate the interpretability of our model through two additional experiments. Firstly, we assess the contribution of individual tabular variables using the SHapley Additive exPlanations (SHAP) tool in Fig. 5. SHAP values demonstrate how each tabular variable impacts the model output, where larger absolute values signal a more significant effect on the classification result. For the MCI conversion prediction task (Fig. 5a), the most discriminative feature is FDG_bl which is a summary measure from FDG-PET. The plot shows that low FDG_bl values have high positive SHAP values, strongly pushing the model to predict a conversion to AD. This aligns with clinical understanding, where reduced glucose metabolism (a low FDG_bl value) is a key biomarker for neurodegeneration and progression from MCI to dementia [43]. For the AD classification task (Fig. 5b), similarly to the MCI model, low ABETA_bl values have strong positive SHAP values, pushing the model to classify the patient as having AD. This is consistent with the amyloid cascade hypothesis, where low amyloid in cerebrospinal fluid (a proxy for high plaque buildup in the brain) is a core hallmark of AD [44].

The APOE4 feature is also highly impactful in AD classification. Having zero APOE4 risk alleles results in large negative SHAP values, strongly pushing the model away from an AD classification. This directly reflects the role of APOE4 as a significant genetic risk factor for the disease. Consistent with the findings reported in [43], preclinical

Table 8
Parameter efficiency analysis for different tuning strategies.

Tuning Strategy	Tunable Parameters	Relative Ratio (%)	Training time (ms)	Inference time (ms)	Inference memory	Training memory	GFLOPs
FT	~ 0.5M	100	1.9 ± 0.3	0.26 ± 0.03	9.72 MB	14.14 MB	0.8042
GPF	~ 5K	1	2.1 ± 0.2	0.28 ± 0.06	10.01 MB	12.94 MB	0.8051
GPF-plus	~ 0.21M	42	2.4 ± 0.2	0.29 ± 0.10	12.91 MB	16.46 MB	0.8365
LoRA	~ 0.1M	20	4.3 ± 0.2	0.35 ± 0.01	11.12 MB	15.72 MB	0.8444
AdapterGNN	~ 0.1M	20	7.9 ± 2.5	1.22 ± 0.04	11.82 MB	17.04 MB	0.8586
PHGNN	~ 0.03M	6	1.8 ± 0.1	0.24 ± 0.04	10.31 MB	12.98 MB	0.8223

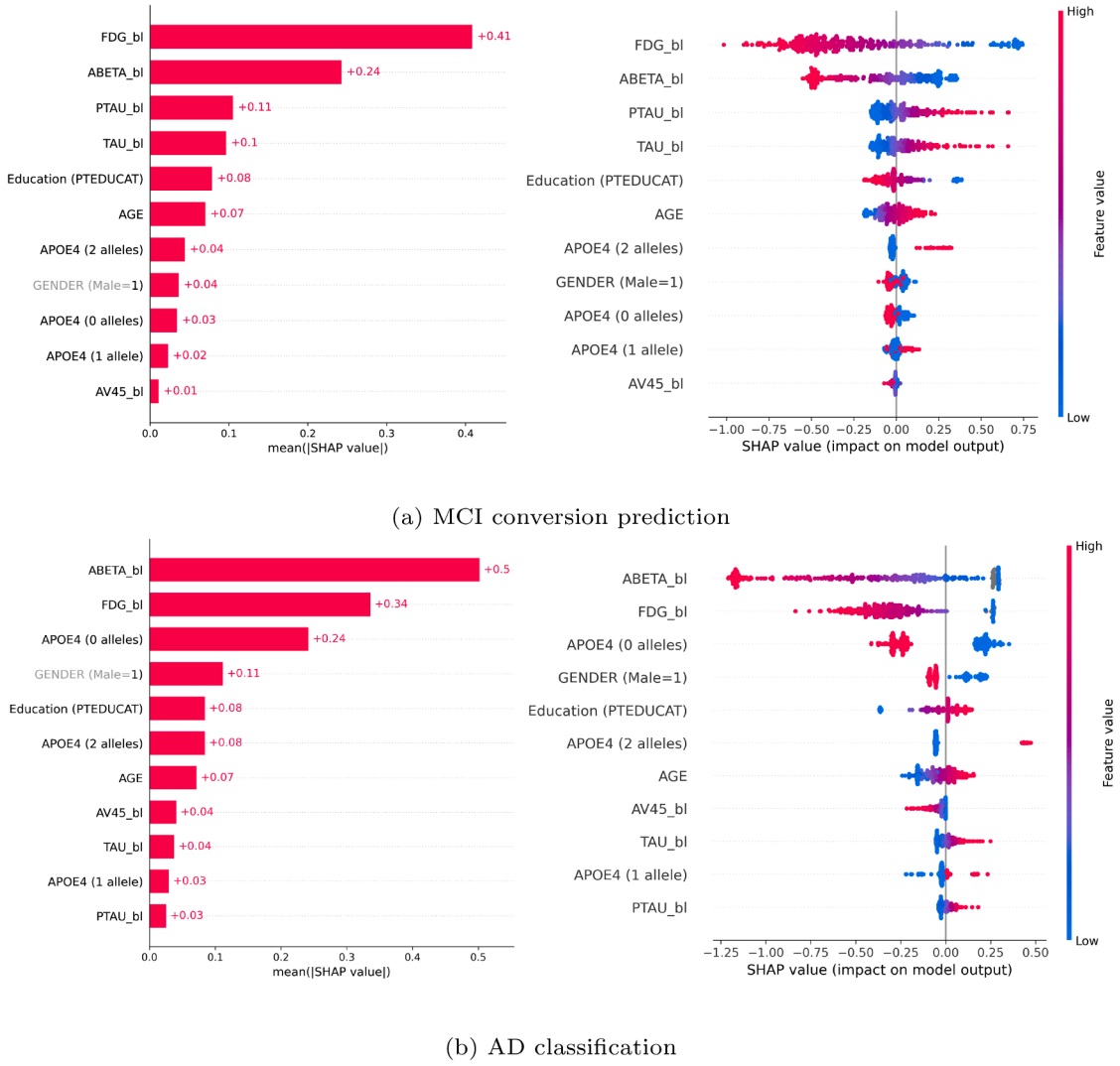


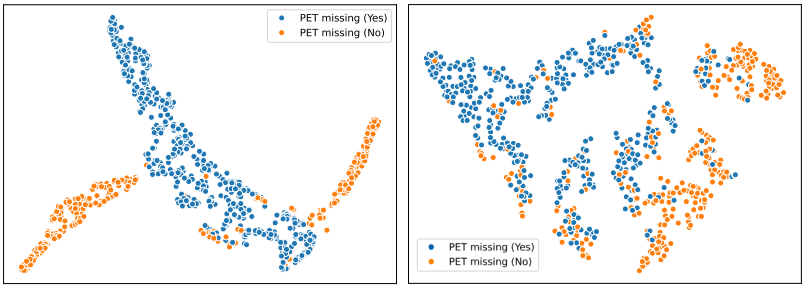
Fig. 5. Visualizations of tabular variables with SHAP values for the (a) MCI conversion prediction task and (b) AD classification task. Within each subfigure, the right part shows the SHAP value distribution, while the left part presents the mean absolute SHAP values. A higher absolute SHAP value signifies a more substantial influence on the classification result.

proof is provided that the existence of the APOE4 allele acts as a risk determinant for AD.

We also visualize the feature embeddings generated by HGNN and PHGNN for ADNI-1 and ADNI-2 where part of the PET modilities are missing, as shown in Fig. 6. In HGNN (Fig. 6a), the absence of PET data creates a null input, leading to distinct clusters with and without PET modality, respectively. The PHGNN model fundamentally solves this problem (Fig. 6b). When PET data is missing, the learnable soft prompts are activated and they dynamically generate a virtual PET feature representation based on the patient other available modalities, such as MRI and tabular data. This synthesized feature acts as an informed substitute, effectively filling the semantic gap. As a result, the model no longer perceives “missing PET" subjects as a distinct group and can process both “real PET" and “missing PET" subject data through a unified information structure. This alignment forces the model to learn deeper, shared biological patterns across all patients, mapping clinically similar individuals to nearby points in the embedding space.

5. Discussion

The proposed framework is underpinned by several architectural choices, the effectiveness of which has been validated by our



(a) HGNN output features (b) PHGNN output features

Fig. 6. Deep feature visualizations on ADNI-1/2 for (a) HGNN and (b) PHGNN. Different colors indicate whether the PET imaging for the sample is missing.

experiments. An alternative to our generative pre-training strategy could have been resorting to constrastive learning instead. The latter, however, heavily relies on constructing valid positive pairs via data augmentation [45]. In medical imaging and clinical tabular data, standard augmentations (e.g., adding Gaussian noise to MRI features or randomly dropping clinical attributes) create biologically implausible samples or even alter the underlying pathological semantics (e.g., artificially effectively “curing" a patient by masking a key atrophy marker). This noise can mislead the contrastive objective. AD involves complex, non-linear interactions between varying biomarkers, therefore by masking a subject features and forcing the model to reconstruct them, HyperGraphMAE is explicitly trained to learn these biological interdependencies. Moreover, in clinical practice, advanced imaging like PET is frequently missing due to cost or invasiveness [46]. A generative pre-training task forces the model to reconstruct missing data, which explicitly trains the model to be robust to incomplete data. Besides, to successfully reconstruct masked features, the model is forced to learn strong, non-linear correlations between different modalities, e.g., using available MRI imaging and clinical data to infer biological in PET imaging. This in turn can provide a crucial clinical and economic advantage for centers lacking expensive imaging facilities.

As a result of this pre-training, our model is able to learn complex biological patterns among the different data modalities. At this point, an alternative strategy could have been to use full fine-tuning instead of the proposed prompt tuning strategy. However, using full fine-tuning when the downstream task labels are noisy or the sample distribution is highly imbalanced (e.g., due to scarcity of MCI conversion instances) may result in fine-tuning gradients that destroy the robust, general-purpose representations obtained during pre-training. For example, there are 115 sMCI subjects but only 13 subjects in ADNI-3. Furthermore, the HGNN model can easily find shortcut solutions by overfitting to spurious correlations. Indeed, as shown in Fig. 6a, the HGNN model may erroneously interpret the presence or absence of the PET modality as a discriminative factor for the downstream task. By freezing the HGNN model, our prompt-based approach protects the pre-trained knowledge and only tunes the prompt tokens. This acts as a powerful form of regularization, reducing the model degrees of freedom and forcing it to find a solution based on the biologically meaningful features learned during pre-training.

5.1. Limitations and future work

Despite the promising results, our work has several limitations:

Sensitivity to hyperparameters selection. The construction of the initial hypergraph relies on the k-NN algorithm, making the model performance sensitive to the choice of the hyperparameter k. As shown in our ablation study, an optimal k is crucial for capturing true underlying biological relationships. Future work could explore ways to automatically infer the optimal k. The selection of the number of prompt tokens $|P|$ also currently relies on manual tuning. This process lacks theoretical guidance and can be inefficient when adapting to new tasks. To address this, we plan to investigate meta-learning optimization approaches that can dynamically determine the optimal prompt configuration for a given task without extensive manual supervision.

Additional modalities and temporal information. While our multimodal approach is robust, it currently relies on FDG-PET images. We also conducted preliminary experiments incorporating Amyloid-PET, which demonstrated promising potential for AD diagnosis. The incorporation of additional PET imaging modalities, such as Tau-PET, would provide a more comprehensive diagnosis. Furthermore, our current framework operates on static, baseline data. Since AD is a progressive neurodegenerative disorder, A significant future direction is to extend PHGNN to leverage longitudinal data.

Scalability issues. While modeling each patient as a node in a hypergraph has many advantages, this transductive setting can face scalability challenges in terms of computational complexity and memory usage when applied to real-world clinical datasets with a very large number of patients. This is generally not the case for inductive models where patients are processed independently.

“Black box" nature of prompt tokens. In Section 4.8 we have assessed the intepretability of our model in terms of the input tabular features. However, the prompt tokens themselves remain abstract vectors lacking a direct clinical or biological meaning. Understanding what biological patterns are encoded in the prompts to compensate for missing modalities during the tuning phase remains an open challenge.

Beyond AD diagnosis. Finally, while this study focuses on AD, we believe that the impact of the prompted hypergraph framework underpinning PHGNN can go beyond the medical application presented in this paper. Indeed, we see our framework as a contribution to the broader field of multimodal graph machine learning providing a general solution for multimodal learning tasks. In future work we plan to showcase this generalizability by extending our model to handle large-scale multi-task multimodal graph machine learning problems. Specifically, we envision a single, frozen, pre-trained PHGNN backbone that can be tailored to several distinct downstream tasks, where each task is defined solely by its corresponding lightweight, task-specific prompt.

6. Conclusion

In this paper we introduced PHGNN, a novel framework for AD diagnosis and MCI conversion prognosis based on hypergraphs and prompt learning. Our framework leverages multiple modalities of imaging and tabular data and can cope with common scenarios in medical applications where labeled data is scarce. To this end, we introduced a novel prompt learning strategy tailored for hypergraphs. Our experiments have shown that this strategy is highly efficient in terms of tunable parameters, running time, and memory consumption while outperforming alternative tuning approaches in terms of downstream task accuracy. The resulting framework has been shown to significantly outperform SOTA multimodal classification

models, including in challenging cross-domain validation scenarios. In addition, the clinical insights derived from these visualizations align well with established findings in the literature. Importantly, the proposed prompted hypergraph paradigm serves as a generalizable methodological contribution to the broader multi-modal graph machine learning field, which will be demostrated in future work.

CRediT authorship contribution statement

Chenyu Liu: Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Luca Cosmo:** Validation, Data curation; **Luca Rossi:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Data availability

The authors do not have permission to share data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This article utilizes data obtained from ADNI, OASIS, and NACC. While the investigators of ADNI, OASIS, and NACC contributed to the design, implementation, and/or collection of these datasets, they were not involved in the analysis or writing of this paper. Further details and a complete list of the subjects can be found at adni.loni.usc.edu, sites.wustl.edu/oasisbrains, and naccdata.org.

References

[1] S. Spasov, L. Passamonti, A. Duggento, P. Lio, N. Toschi, A.D.N. Initiative, et al., A parameter-efficient deep learning approach to predict conversion from mild cognitive impairment to Alzheimer's disease, Neuroimage 189 (2019) 276–287.
[2] S. Pölsterl, T.N. Wolf, C. Wachinger, Combining 3D image and tabular data via the dynamic affine feature map transform, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V 24, Springer, 2021, pp. 688–698.
[3] X. Gao, F. Shi, D. Shen, M. Liu, Task-induced pyramid and attention GAN for multimodal brain image imputation and classification in Alzheimer's disease, IEEE J. Biomed hHealth Inf. 26 (1) (2021) 36–43.
[4] Y. Zhang, K. Sun, Y. Liu, F. Xie, Q. Guo, D. Shen, A modality-flexible framework for Alzheimer's disease diagnosis following clinical routine, IEEE J. Biomed. Health Inf. 29 (1) (2025) 535–546.
[5] B. Lei, Y. Zhu, S. Yu, H. Hu, Y. Xu, G. Yue, T. Wang, C. Zhao, S. Chen, P. Yang, et al., Multi-scale enhanced graph convolutional network for mild cognitive impairment detection, Pattern Recognit. 134 (2023) 109106.
[6] A.I. Aviles-Rivero, C. Runkel, N. Papadakis, Z. Kourtzi, C.-B. Schönlieb, Multi-modal hypergraph diffusion network with dual prior for Alzheimer classification, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2022, pp. 717–727.
[7] J. Xu, C. Yuan, X. Ma, H. Shang, X. Shi, X. Zhu, Interpretable medical deep framework by logits-constraint attention guiding graph-based multi-scale fusion for Alzheimer's disease analysis, Pattern Recognit. 152 (2024) 110450.
[8] D. Cai, M. Song, C. Sun, B. Zhang, S. Hong, H. Li, Hypergraph structure learning for hypergraph neural networks, in: Proceedings of IJCAI, 2022, pp. 1923–1929.
[9] J. Liao, J. Yan, Q. Tao, E. Zhang, Y. Zhang, A novel hypergraph neural network combining multi-view learning with density awareness, Pattern Recognit. (2025) 111775.
[10] B. Min, H. Ross, E. Sulem, A.P.B. Veyseh, T.H. Nguyen, O. Sainz, E. Agirre, I. Heintz, D. Roth, Recent advances in natural language processing via large pre-trained language models: a survey, ACM Comput. Surv. 56 (2) (2023) 1–40.
[11] S. Cai, X. Liu, J. Yuan, Q. Zhou, Prompt-Ladder: memory-efficient prompt tuning for vision-language models on edge devices, Pattern Recognit. 163 (2025) 111460.
[12] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2009) 1345–1359.
[13] Semi-supervised classification with graph convolutional networks, in: Proceedings of the 5th International Conference on Learning Representations, ICLR '17, 2017.
[14] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks,(2017). arXiv:1710.10903
[15] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, NeurIPS 30 (2017) 1024–1034.
[16] S. Kim, S.Y. Lee, Y. Gao, A. Antelmi, M. Polato, K. Shin, A survey on hypergraph neural networks: an in-depth and step-by-step guide, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6534–6544.
[17] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 33, 2019, pp. 3558–3565.
[18] K.M. Saifuddin, B. Bumgardner, F. Tanvir, E. Akbas, HyGNN: drug-drug interaction prediction via hypergraph neural network, in: 2023 IEEE 39th International Conference on Data Engineering (ICDE), IEEE, 2023, pp. 1503–1516.
[19] X. Li, Y. Zhang, Y. Wang, Z. Liu, Graph convolutional networks for Alzheimer's disease diagnosis, in: Proceedings of IEEE CVPR, 2018.
[20] Y. Zhang, Z. Wang, H. Li, Z. Wei, Multi-view graph neural networks for Alzheimer's disease diagnosis, IEEE Trans. Med. Imaging 40 (3) (2021) 645–656.
[21] D. Shen, L. Wang, Y. Liu, Dynamic graph neural networks for predicting Alzheimer's disease progression, Neuroimage 216 (2022) 116933.
[22] T.B. Brown, Language models are few-shot learners,(2020). arXiv:2005.14165
[23] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners,(2020). arXiv:2012.15723
[24] K. Zhou, J. Yang, C.C. Loy, Z. Liu, Learning to prompt for vision-language models, IJCV 130 (9) (2022) 2337–2348.
[25] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, S.-N. Lim, Visual prompt tuning, in: ECCV, Springer, 2022, pp. 709–727.
[26] T. Fang, Y. Zhang, Y. Yang, C. Wang, L. Chen, Universal prompt tuning for graph neural networks, NeurIPS 36 (2023) 55315–55328.
[27] X. Sun, J. Zhang, X. Wu, H. Cheng, Y. Xiong, J. Li, Graph prompt learning: a comprehensive survey and beyond,(2023). arXiv:2311.16534
[28] Y. Zhu, J. Guo, S. Tang, SGL-PT: a strong graph learner with graph prompt tuning,(2023). arXiv:2302.12449
[29] Y. Gao, Y. Feng, S. Ji, R. Ji, HGNN+: general hypergraph neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 45 (3) (2022) 3181–3199.
[30] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, J. Tang, Graphmae: self-supervised masked graph autoencoders, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 594–604.
[31] C.R. Jack, Jr, M.A. Bernstein, N.C. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P.J. Britson, L.W. Jennifer, C. Ward, et al., The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods, J. Magn. Reson. Imaging: Off. J. Int. Soc. Magn. Reson. Med. 27 (4) (2008) 685–691.
[32] P.J. LaMontagne, T.L.S. Benzinger, J.C. Morris, S. Keefe, R. Hornbeck, C. Xiong, E. Grant, J. Hassenstab, K. Moulder, A.G. Vlassenko, et al., OASIS-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and Alzheimer disease, medRxiv (2019) https://doi.org/10.1101/2019.12.13.19014902.
[33] D.L. Beekly, E.M. Ramos, G. van Belle, W. Deitrich, A.D. Clark, M.E. Jacka, W.A. Kukull, et al., The national Alzheimer's coordinating center (NACC) database: an Alzheimer disease database, Alzheimer Dis. Assoc. Disord. 18 (4) (2004) 270–277.
[34] H. Xu, J. Wang, Q. Feng, Y. Zhang, Z. Ning, Domain-specific information preservation for Alzheimer's disease diagnosis with incomplete multi-modality neuroimages, Med. Image Anal. 101 (2025) 103448.
[35] S. Parisot, S.I. Ktena, E. Ferrante, M. Lee, R. Guerrero, B. Glocker, D. Rueckert, Disease prediction using graph convolutional networks: application to autism spectrum disorder and Alzheimer's disease, Med. Image Anal. 48 (2018) 117–130.
[36] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al., Lora: low-rank adaptation of large language models, ICLR 1 (2) (2022) 3.
[37] S. Li, X. Han, J. Bai, AdapterGNN: parameter-efficient fine-tuning improves generalization in GNNs, in: Proceedings of the AAAI Conference on Artificial Intelligence, 38, 2024, pp. 13600–13608.
[38] J. Xia, L. Wu, J. Chen, B. Hu, S.Z. Li, Simgrace: a simple framework for graph contrastive learning without data augmentation, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 1070–1079.
[39] P. Veličković, W. Fedus, W.L. Hamilton, P. Liò, Y. Bengio, R.D. Hjelm, Deep graph infomax,(2018). arXiv:1809.10341
[40] T. Wei, Y. You, T. Chen, Y. Shen, J. He, Z. Wang, Augmentations in hypergraph contrastive learning: fabricated and generative, NeurIPS 35 (2022) 1909–1922.
[41] X. Wu, K. Zhou, M. Sun, X. Wang, N. Liu, A survey of graph prompting methods: techniques, applications, and challenges,(2023). arXiv:2303.07275
[42] J. Samper-González, N. Burgos, S. Bottani, S. Fontanella, P. Lu, A. Marcoux, A. Routier, J. Guillon, M. Bacci, J. Wen, et al., Reproducible evaluation of classification methods in Alzheimer's disease: framework and application to MRI and PET data, Neuroimage 183 (2018) 504–521.
[43] E.M. Reiman, R.J. Caselli, L.S. Yun, K. Chen, D. Bandy, S. Minoshima, S.N. Thibodeau, D. Osborne, Preclinical evidence of Alzheimer's disease in persons homozygous for the ε4 allele for apolipoprotein E, N. Engl. J. Med. 334 (12) (1996) 752–758.
[44] J.A. Hardy, G.A. Higgins, Alzheimer's disease: the amyloid cascade hypothesis, Science 256 (5054) (1992) 184–185.
[45] T. Zhao, Y. Wang, S. Xu, T. Yang, J. Gao, J. Guo, Dual-level noise augmentation for graph clustering with triplet-wise contrastive learning, Pattern Recognit. (2025) 112463.
[46] W. Xiong, T. Wang, X. Chen, Y. Zhang, W. Zhang, Q. Feng, M. Huang, A.D.N. Initiative, et al., Disentanglement and codebook learning-induced feature match network to diagnose neurodegenerative diseases on incomplete multimodal data, Pattern Recognit. 165 (2025) 111597.