



## OPEN Alzheimer's disease recognition using graph neural network by leveraging image-text similarity from vision language model

Byounghwa Lee✉, Jeong-Uk Bang, Hwa Jeon Song & Byung Ok Kang

Alzheimer's disease (AD), a progressive neurodegenerative condition, notably impacts cognitive functions and daily activity. One method of detecting dementia involves a task where participants describe a given picture, and extensive research has been conducted using the participants' speech and transcribed text. However, very few studies have explored the modality of the image itself. In this work, we propose a method that predicts dementia automatically by representing the relationship between images and texts as a graph. First, we transcribe the participants' speech into text using an automatic speech recognition system. Then, we employ a vision language model to represent the relationship between the parts of the image and the corresponding descriptive sentences as a bipartite graph. Finally, we use a graph convolutional network (GCN), considering each subject as an individual graph, to classify AD patients through a graph-level classification task. In experiments conducted on the ADReSSo Challenge datasets, our model surpassed the existing state-of-the-art performance by achieving an accuracy of 88.73%. Additionally, ablation studies that removed the relationship between images and texts demonstrated the critical role of graphs in improving performance. Furthermore, by utilizing the sentence representations learned through the GCN, we identified the sentences and keywords critical for AD classification.

**Keywords** Alzheimer's disease, Bipartite graph, Dementia, Multimodal, Graph neural network, Vision language model

Alzheimer's disease (AD), a progressive neurodegenerative condition, significantly affects cognitive functions and the ability to perform daily activities, primarily impairing memory. The progression of AD underlines the importance of early detection and intervention, which can potentially slow the cognitive decline associated with the disease. Consequently, research initiatives are increasingly focused on the development of diagnostic tools, including the analysis of speech and transcribed text, to identify AD in its early stages.

Conventional AD screening methods, such as the Mini Mental Status Examination (MMSE)<sup>1</sup> and the Montreal Cognitive Assessment (MoCA)<sup>2</sup>, rely on the subjective judgments of clinicians, which may result in errors and high inter-rater variability<sup>3</sup>. To address these issues, there are numerous studies on automatically assessing AD<sup>4,5</sup>. One promising approach for AD screening involves utilizing speech signals<sup>4</sup>. Speech signals provide the benefit of being naturally and effortlessly gathered continuously over the day, thereby enabling the accumulation of substantial data volumes without overburdening the participants or researchers<sup>6</sup>.

A leading approach in speech-based AD detection is the *Cookie Theft* picture description task<sup>7</sup>, which is one of the tasks from the Boston Diagnostic Aphasia Examination (BDAE)<sup>8</sup>. In this task, participants are asked to describe everything they see in a picture using spontaneous speech, and the responses are recorded as audio files. The notable dataset associated with this task is the Pitt corpus<sup>7</sup>. The ADReSS Challenge dataset<sup>9</sup> and the ADReSSo Challenge dataset<sup>10</sup>, both subsets of the Pitt corpus and matched for age and gender, are utilized as benchmark datasets for AD detection. The ADReSS Challenge dataset provides human-transcribed texts along with the audio files, which does not align with the objective of diagnosing dementia automatically without human intervention. Hence, this study employs the ADReSSo Challenge dataset, which offers purely audio files, to classify two health status: Healthy Control (HC) and AD.

Previous studies on AD classification based on picture description have typically involved three types of models: (1) audio-only models<sup>11,12</sup>, which extract acoustic features from speech signals using either traditional

Integrated Intelligence Research Section, Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea. ✉email: byounghwa.lee@etri.re.kr

methods or deep embeddings, (2) text-only models<sup>13,14</sup>, which convert audio into text and utilize pre-trained language models such as BERT<sup>15</sup> to extract features, and (3) multimodal models<sup>16–18</sup>, which utilize both audio and textual modalities, either by fusing their features or employing attention mechanisms during training phase. Recently, there has been a study leveraging feedback from Large Language Model (LLM) as a feature to improve performance<sup>19,20</sup>.

Our model introduces a method that incorporates image modality as well as text modality to automatically predict dementia by representing the relationships between images and texts as a graph. The spontaneous speech of the participants is transcribed into text using the Whisper<sup>21</sup>, a representative automatic speech recognition (ASR) system. Subsequently, a vision language model (VLM), specifically Bootstrapping Language-Image Pre-training (BLIP)<sup>22</sup>, is utilized to depict the connections between different sections of an image and their respective descriptive sentences in the form of a bipartite graph. In the final step, a graph convolutional network (GCN) is employed to perform graph-level classification for classification of AD patients, treating each subject as a separate graph.

The main contributions of this paper are summarized as follows:

- This work is the first to utilize a graph for modeling the image-text relationship with a vision language model (VLM) in the domain of dementia detection.
- The bipartite graph encapsulates the image-text relationship, which is crucial for performance enhancement. The image-text relations between the AD and HC groups are structurally distinct enough to enable differentiation by a standard GCN.
- Our model achieved superior AD classification performance on the ADReSSo Challenge dataset compared to existing state-of-the-art (SOTA) models, attaining an accuracy of 88.73%, which surpasses the previously highest recorded SOTA model accuracy of 87.32%<sup>19</sup>. The peak accuracy achieved during one of the runs was 91.55%.
- In explainability experiments using embedding vectors from a trained graph neural network, we identified key sentences and keywords essential for AD classification by comparing pooled embeddings of AD and HC groups with embeddings of individual sentences.
- By employing a graph-based approach and avoiding computationally intensive Transformers, the model not only simplifies its architecture but also significantly reduces computational costs and improves memory efficiency. Our proposed model is capable of automatically recognizing AD from audio transcription to classification, without relying on human-derived handcrafted features. Given any audio file from a picture description task, the model's performance is reproducible with only minor adjustments to hyperparameters.

## Related work

### Dementia detection

In this section, we primarily introduce studies focused on dementia detection using the ADReSSo Challenge dataset.

#### *Unimodal model*

The most natural approach to using spontaneous speech data involves directly feeding the audio modality into the model. One study<sup>11</sup> combined traditional acoustic features with acoustic embeddings from wav2vec 2.0<sup>23</sup>, subsequently employing a support vector machine (SVM) for AD classification, resulting in an accuracy of 67.6%. Another study<sup>12</sup> employed various pre-trained audio models, including wav2vec 2.0, to acquire acoustic embeddings and applied deep learning approaches for AD classification, achieving an accuracy of 78.9%.

In general, transcribing audio signals into text and then training models on the textual modality tends to yield superior performance compared to solely utilizing the audio modality. One study<sup>13</sup> combined the final three states of the pre-trained BERT sequence classifier with the confidence score input produced by the ASR system, resulting in an accuracy of 84.51%. In addition to deep textual embeddings from BERT, another study<sup>14</sup> utilized a set of handcrafted features (including syntactic, readability, and lexical diversity) alongside preprocessing steps that integrated silence segments. They trained Logistic Regression (LR) and SVM classifiers, achieving an accuracy of 84.51%.

#### *Multimodal model*

The most common multimodal approach involves considering both the audio and the text modalities simultaneously. In this approach, the audio modality can be utilized either as it is or processed into an image domain through a log-Mel spectrogram. One study<sup>24</sup> integrated various acoustic features, including x-vectors, prosody, and emotional embeddings, along with word embeddings, resulting in an accuracy of 80.30%. Another study<sup>25</sup> introduced the WavBERT model, which involved converting the wav2vec output into the BERT input to retain non-semantic information. They also incorporated sentence-level pauses into ASR transcripts, leading to an accuracy of 83.10%.

In another research<sup>16</sup>, global fusion combining BERT with several acoustic models, such as x-vectors and encoder-decoder ASR embeddings, yielded an accuracy of 84.51%. Meanwhile, one study<sup>17</sup> leveraged full transcripts as prompts to enhance speech segment training, addressing the limited perspective of Whisper due to the constrained audio segment lengths during fine-tuning. They achieved an accuracy of 84.51%. The research presented in<sup>18</sup> introduced a multimodal model incorporating Co-attention, Deep Context, and label smoothing techniques. Co-attention enables simultaneous consideration of different representations, Deep Context captures both low- and high-level syntactic and semantic information, and label smoothing prevents overconfidence. Texts were encoded using BERT, while audio signals were transformed into log-Mel spectrograms and fed into Data-efficient image Transformers (DeiT)<sup>26</sup>. They achieved an accuracy of 85.35%. In another study<sup>19</sup>, feedback

from ChatGPT was treated as an opinion feature, concatenated with text embedding from BERT and audio embedding from wav2vec 2.0, achieving an accuracy of 87.32%. A study utilizing another LLM, such as Mistral 7B, on the ADReSS dataset achieved an accuracy of 81.3%<sup>20</sup>.

One study<sup>27</sup> leveraged both images and descriptive texts, utilizing insights from extensively pre-trained image-text alignment models, particularly Contrastive Language-Image Pre-training (CLIP)<sup>28</sup>, to enhance accuracy. However, their research differs from ours in that they do not examine the relationship between images and texts using a graph.

Another study<sup>29</sup> introduced a tensor fusion layer to integrate transcribed text, audio, and log-Mel spectrograms, achieving an accuracy of 86.25% on the ADReSS dataset. Additionally, a separate study<sup>30</sup> employed Neural Architecture Search to propose an optimal CNN structure and presented a novel approach to integrating text and log-Mel spectrogram modalities, resulting in an accuracy of 92.08% on the ADReSS dataset. Furthermore, another study<sup>31</sup> utilized audio, lexical, and disfluency features, combining them through LSTM and a gating mechanism, achieving an accuracy of 79.2% on the ADReSS dataset.

## Background

### *Vision language model*

Vision language models (VLMs) enhance downstream vision and language tasks by pre-training on large image-text pairs datasets. CLIP employs contrastive learning to match text and image embeddings by selecting the most similar pair, while ALign the image and text representations Before Fusing (ALBEF)<sup>32</sup> aligns unimodal representations before fusing them into a multimodal encoder with assistance from momentum distillation. Though CLIP and ALBEF use web image-text pairs for pre-training, the noisy data isn't ideal for learning. BLIP improves this by using Captioning and Filtering (CapFilt) and Multimodal mixture of Encoder-Decoder (MED). BLIP-2<sup>33</sup> builds on BLIP with a more computationally efficient approach. To leverage the unique functionality of BLIP, we selected BLIP over BLIP-2 for image-text embedding and similarity measurement.

### *Graph convolutional network*

Graph neural networks (GNNs)<sup>34</sup> are designed to process graph data, similar to how convolutional neural networks (CNNs) process adjacent pixels in images. In GNNs, node information is exchanged between neighboring nodes through message passing to update embeddings. GCN<sup>35</sup>, a type of GNN, applies convolutional operations to graphs, aggregating information from neighboring nodes. In this study, we use the following GCN model:

$$x_i^l = W_1^l x_i^{l-1} + W_2^l \sum_{j \in \mathcal{N}(i)} e_{j,i} \cdot x_j^{l-1}, \quad (1)$$

where  $x_i^l$  and  $x_i^{l-1}$  are the node embedding vectors for the node  $i$  of the  $l$ -th layer and the  $(l-1)$ -th layer, respectively. During the training phase, node  $i$  can represent either an image or a text node. If node  $i$  is an image node, then node  $j$  must be a text node, and vice versa. The edge weight is denoted as  $e_{j,i}$  from source node  $j$  to target node  $i$ .  $\mathcal{N}(i)$  is the set of neighboring nodes of node  $i$ , and  $W_1^l$  and  $W_2^l$  are learnable parameters.

## Methods

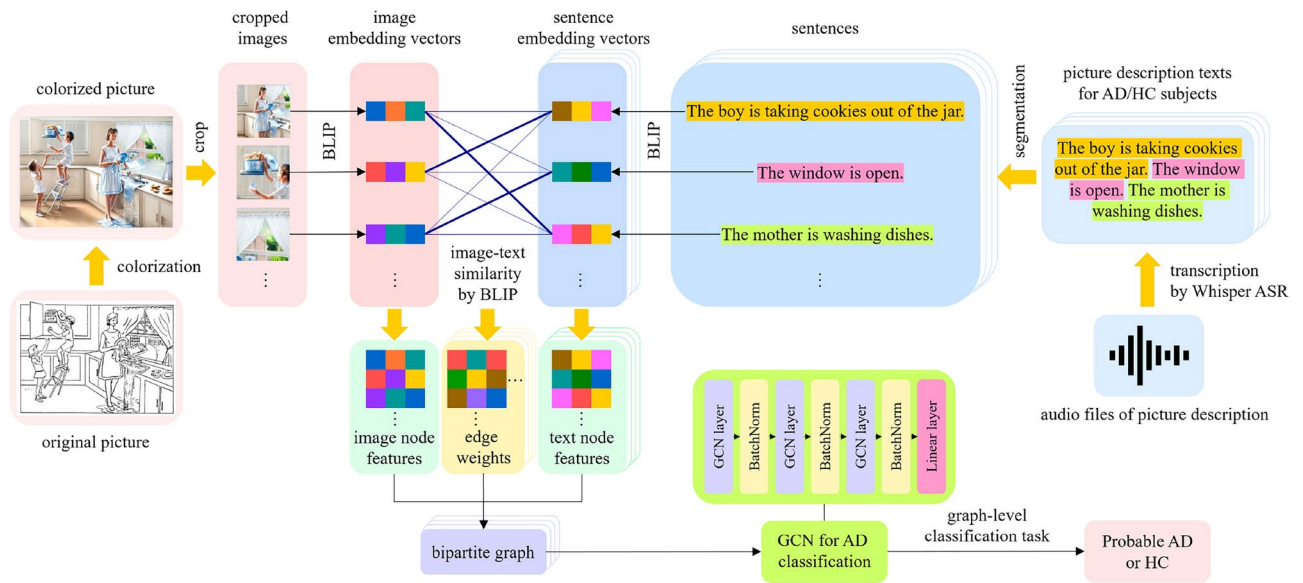
Figure 1 illustrates the overall framework of our model. The framework consists of four main components: 1) image node processing (in red), 2) sentence node processing (in blue), 3) bipartite graph construction (in purple), which includes image-text similarity based on a VLM, and 4) graph convolutional network (in green) and AD classification.

### Image node processing

**Colorization** The BLIP model is optimized for color images as it is pre-trained on the COCO dataset<sup>36</sup>. Therefore, using grayscale cartoon-style images such as the *Cookie Theft* picture (shown in Fig. 2a) directly for feature extraction is not ideal. When grayscale images are fed into BLIP as input, there are quite a few instances where inappropriate captions are generated. Hence, we colorized the *Cookie Theft* image using a generative AI tool. The effectiveness of colorization is discussed in Table 4.

**Crop** Subsequently, we cropped the image into 10 square-sized sub-images, as depicted in Fig. 2c. Each sub-image was cropped to varying sizes and then resized to the same sizes afterward. Each cropped image represents a distinct scenario. For instance, one image depicts water overflowing from a sink, while another portrays a boy stealing cookies. To accommodate descriptions depicting the overall context, cropped images close to the full size were also included.

Since the *Cookie Theft* picture is the standard image for AD recognition, several studies have manually divided it into halves or quadrants<sup>37-39</sup> to analyze and classify AD groups. Another study manually divided the picture into 10 cropped areas based on words as a seed<sup>40</sup>. Some studies employ eye-tracking techniques to obtain areas of interest (AOI)<sup>41,42</sup>. In addition to these methods, we identified important areas in the picture by analyzing the relationship between frequently uttered words and the picture using Grad-CAM<sup>43</sup> for the VLM in Fig. 2b. Based on the aforementioned references and Grad-CAM results, we meticulously cropped the picture into 10 sub-images.



**Fig. 1.** Framework of the bipartite graph neural network model for AD classification. The image modality (in red) and text modality (in blue) are connected through bipartite graphs (in violet) based on the vision language model. Here, image-text similarity, inspired by VLM, is the edge weight of the bipartite graphs.



**Fig. 2.** (a) The original *Cookie Theft* picture. (b) Heatmap generated by Grad-CAM for the VLM. (c) The image cropping process for the colorized *Cookie Theft* picture.

**Embeddings** Following that, we utilized BLIP to extract the embedding vector for each cropped image. Here, the image embedding vector is obtained considering the corresponding sentence. In other words, even for the same cropped image, the value of the embedding vector varies depending on the corresponding sentence. Then, for each sample, we average the image embedding vectors across all sentences to obtain the final cropped image embedding vector. Each cropped image yields one embedding vector, which serves as an attribute of the image node in the bipartite graph. The practical implementation details for image embeddings are provided in the Experiments section.

**Sentence node processing**

**Transcription** We transcribed the given speech signals into text using the prominent ASR system, Whisper-large<sup>21</sup>. This process yielded one text file per subject.

**Segmentation** Subsequently, to facilitate matching with cropped images, we segmented the entire text into individual sentences, using punctuation marks such as periods, exclamation points, and question marks as delimiters.

**Embeddings** Following this, we employed BLIP to extract the embedding vector for each sentence. Here, the sentence embedding vector is obtained considering the corresponding cropped image. In other words, even for the same sentence, the value of the embedding vector varies depending on the corresponding cropped image. Then, for each sample, we averaged the sentence embedding vectors across all cropped images to obtain the final sentence embedding vector. Each sentence yields one embedding vector, which serves as an attribute of the text node in the bipartite graph. The practical implementation details for sentence embeddings are presented in the Experiments section.

**Bipartite graph construction: image-text similarity based on vision language model**

We employ BLIP to extract the cosine similarity between the cropped images and sentences. A well-describing sentence yields a high cosine similarity. Upon examining several sample sentences, we confirmed that BLIP



exhibits high relevance in computing image-text similarity. In contrast to research utilizing CLIP, we chose BLIP due to its superior performance. Furthermore, CLIP tend to focus on single-word aspects, which are less aligned with our research direction.

The image-text cosine similarity is computed for all pairs of cropped images and sentences, subsequently becoming the edge weights of the bipartite graph. The cosine similarity matrix essentially serves as the adjacency matrix of the graph. This approach enables the establishment of informative and reliable connectivity between the image and text modalities.

The nodes of the bipartite graph consist of image nodes, with embedding vectors of the cropped images as attributes, and text nodes, with embedding vectors of the sentences as attributes. Each subject corresponds to one bipartite graph. The number of image nodes is fixed at 10, while the number of text nodes corresponds to the number of sentences in the text. The practical implementation details for calculation of image-text similarity are provided in the Experiments section.

The bipartite graph captures the complex relational information between participants' spoken descriptions and sub-images, encompassing several crucial aspects. Firstly, when participants thoroughly describe each part of the picture, the corresponding sentence associated with the sub-image receives a large edge weight. Secondly, comprehensive descriptions of all situations within the picture are essential; thus, if a participant provides an all-encompassing description, all image nodes will have at least one large edge weight value, connecting to the corresponding related sentence node. Thirdly, utterances unrelated to the image will not have large edge weight values with any image node. The bipartite graph encapsulates various implicit pieces of information, including the aforementioned aspects, which could significantly enhance performance if leveraged properly. If only the purely textual (or audio) modality is considered, there is a limitation in that such cases cannot be included in the model training.

Figure 3 presents examples of bipartite graphs for the AD and HC group samples, illustrating the different patterns of edge weights between the two samples. Since the edge weights represent the image-text similarity in the VLM, it is evident that, in the HC group samples, each text is more broadly connected to various crop images. The practical implementation details for constructing a bipartite graph dataset compatible with GNNs are presented in the Experiments section.

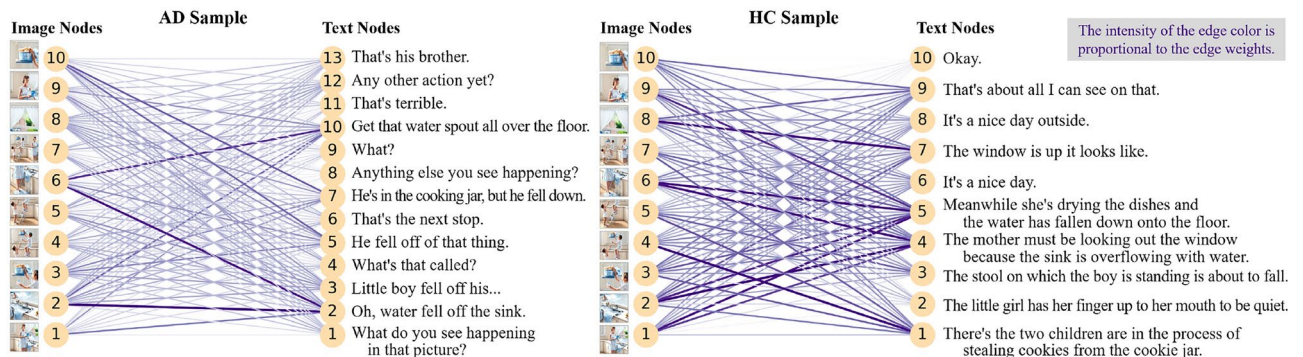
### Graph convolutional network and AD classification

#### Graph convolutional network

Using GCN, the embedding vectors of image and text nodes can be updated by incorporating the connectivity information from neighboring nodes. In this process, information from neighbors is reflected proportionally to the edge weights. In our case, the edge weights are determined by the cosine similarity between the cropped images and sentences. Hence, the GCN model updates each node's information more prominently when the relationship between images and texts is closer.

We employ three GCN layers; therefore,  $l$  can take on the values of 1, 2, or 3. When  $l = 1$ ,  $x_i^{l-1}$  represents the initial embedding acquired through BLIP. At the final layer ( $l = L$ ), we can obtain final node embeddings  $x_i^L$  for all nodes  $i$ , where  $L$  is 3 for our case. Here, we denote the final image and text node embeddings as  $h_{s,i}^v$  and  $h_{s,j}^t$ , respectively, where  $s$  is the subject index,  $i$  is the cropped image index, and  $j$  is the sentence index.

Through a GCN that considers edge weights, the relationships between sub-images and sentences are learned by accounting for local structural information up to 3 hops. The final embedding of each node reflects information from neighboring image nodes and neighboring text nodes. For instance, in the case of the *Cookie Theft* picture, consider the sub-image where a boy is standing on a stool to grab a cookie while a girl reaches out beside him. Without using a GCN, all of these actions would need to be captured in a single sentence for the model to learn from this sub-image. However, by using a GCN, the utterance describing the boy standing on the stool, the utterance about the boy grabbing the cookie, and the utterance about the girl reaching out can all be incorporated as neighboring text nodes in the learning process, making generalization more feasible.



**Fig. 3.** Examples of bipartite graphs for AD and HC group samples, respectively. The darker edges represent higher edge weights. In the AD sample, the edges (image node, text node) = (2, 2), (6, 2), (10, 7), (2, 10), (6, 10) have higher weights, indicating a tendency for the text to focus on specific crop images. In contrast, in the HC sample, the edges (image node, text node) = (4, 1), (5, 3), (2, 4), (6, 4), (1, 5), (2, 5), (6, 5), (9, 5), (8, 7) have relatively higher weights, showing that the text is more evenly connected across multiple crop images.

### Graph-level classification

Once we have obtained the final embedding vectors for all nodes, we need to aggregate them into a graph-level embedding vector for the graph-level classification task. We employed mean pooling, where the graph-level embedding vector is the mean of all node embedding vectors. The global mean pooling vector for all nodes and for subject  $s$ , denoted as  $h_s$ , is defined as

$$h_s = \frac{1}{2}(\langle h_{s,i}^v \rangle + \langle h_{s,j}^t \rangle), \quad (2)$$

where  $\langle \cdot \rangle$  denotes the mean operation, and  $h_{s,i}^v$  and  $h_{s,j}^t$  represent the final image and text node embedding vectors, respectively. According to (2), regardless of the number of sentences for each sample, texts are pooled with equal weight to images.

Then, the pooled vector  $h_s$  is fed into the Linear layer for classification of AD versus HC as follows:

$$y = \text{Linear}(h_s). \quad (3)$$

The training and validation loss are calculated using cross entropy. As a result, our proposed model predicts whether a given data sample belongs to AD or HC. Detailed practical information is provided in Experiments section.

## Dataset

The basic statistics of the datasets, including the number of sentences per one sample and the average number of words in one sentence, are shown in Table 1. The ADReSSo Challenge dataset is designed for three tasks<sup>10</sup>, but we only utilize the audio files and ground truth labels pertaining to the *Cookie Theft* picture description task. The ADReSSo Challenge dataset, a benchmark dataset for AD detection, is a subset of the Pitt corpus, matched for age and gender; While the Pitt corpus comprises 548 samples, the ADReSSo Challenge dataset consists of 237 samples. The ADReSSo Challenge dataset was carefully constructed by considering the age and gender distribution when dividing the training and test sets, thus reducing the potential bias due to participants' demographics.

The Pitt corpus does not have a standardized train-test split, meaning that performance can vary across studies depending on the samples included in the training or test sets. As a result, accuracy comparisons between different models may not be reliable. Therefore, the Pitt corpus was primarily used to support the ablation study on the ADReSSo challenge dataset.

Although dependent on the transcription results from Whisper, we conducted two primary analyses. Firstly, the average number of sentences per participant's utterance is higher in the AD group for the ADReSSo Challenge dataset, whereas for the Pitt corpus, the two groups are comparable. The higher sentence count in the AD group is attributed to the prevalence of short sentences such as *Okay*, *Yeah*, *Uh-huh*, and similar expressions. This is evident when examining the average word count per sentence. In both datasets, the HC group exhibits a higher average word count per sentence, indicating that participants in the HC group tend to articulate sentences with more words. The detailed train-test split for the Pitt corpus is outlined in the Experiments section.

## Experiments

### Implementation of graph data

#### Image and text processing using LAVIS

We utilize LAVIS<sup>44</sup>, a Python library that includes a wide range of VLMs, for two key processes: 1) the embedding of images and texts, and 2) the calculation of image-text similarity. LAVIS provides access to over ten image-text tasks and more than thirty pretrained weights from SOTA foundation VLMs, including CLIP<sup>28</sup>, ALBEF<sup>32</sup>, BLIP<sup>22</sup>, and BLIP-2<sup>33</sup>.

	ADReSSo				Pitt corpus		
	Train		Test		HC	AD	
	HC	AD	HC	AD			
# Samples	166		71		548 ]		
	79	87	36	35	243	305	
# Sentences	Mean	14.67	20.07	13.44	17.71	14.02	13.45
	Std	7.55	16.18	7.06	8.96	5.61	6.30
# Avg. words	Mean	13.48	10.40	12.89	8.22	8.39	8.03
	Std	13.55	12.77	10.68	4.60	2.45	2.41

**Table 1.** Dataset description. *Std* standard deviation

Specifically, we utilize the [CLS] token from the `blip_feature_extractor` with the base model type, which consists of pretrained weights from the CapFit by the BLIP large model, to acquire image and text embeddings. For text embeddings, we first extract 10 embedding vectors per sentence, each consisting of 768 dimensions, considering the relationships with 10 cropped images corresponding to each sentence. Similarly, for image embeddings, we first extract embedding vectors for each sentence, with the number of vectors being equal to the number of sentences, and each vector comprising 768 dimensions, considering the relationships with sentences corresponding to each cropped image. We then average all image embedding tensors to obtain the final text embeddings, and vice versa for image embeddings.

The cosine similarity between images and texts is calculated using `blip_image_text_matching` with the base model type, which is fine-tuned with BLIP retrieval weights on the COCO dataset<sup>35</sup>. For each subject, we compute the cosine similarity between all pairs of cropped images  $N_i$  and sentences  $N_t$ , yielding a similarity matrix of size  $N_i \times N_t$ , which serves as the adjacency matrix from a graph perspective.

#### Construction of bipartite graph using PyG

PyG (PyTorch Geometric)<sup>45</sup> is a library built upon PyTorch, designed to seamlessly manage GNNs for diverse applications involving structured data. The `HeteroData` object in PyG describes a heterogeneous graph, holding multiple node and edge types. When certain constraints are applied (the types of nodes are two, and edges are only possible between nodes of different types), a heterogeneous graph can be transformed into a bipartite graph. Thus, we utilize the `HeteroData` type to construct the bipartite graph dataset.

Then, we utilize the `to_hetero` module in PyG to transform a homogeneous GNN model into its heterogeneous counterpart. Subsequently, we opt for the `GraphConv` GNN operator<sup>46</sup>, which accounts for edge weights.

#### Graph neural network structure

Based on the experiments examining the dependency on the number of GCN layers shown in the Results section, the number of GCN layers is determined to be 3. After each GCN layer, batch normalization is applied. Following three GCN layers and three times of batch normalization, a linear layer for binary classification of AD versus HC follows.

We compare two configurations: (1) *light* GCN is a scenario where hidden dimensions diminish by half iteratively, while (2) *full* GCN indicates a scenario where hidden dimensions remain constant, regardless of the increasing number of layers. For the light GCN structure, with a 256-dimensional case, it initially receives features of 768 dimensions. Specifically, at  $L = 1$ , it transforms from 768 to 256 dimensions, at  $L = 2$  from 256 to 128, and so forth, until  $L = 5$  where it reduces from 32 to 16 dimensions. Conversely, in the case of the full GCN structure, for the 256-dimensional scenario, after reducing from 768 to 256 dimensions at  $L = 1$ , the dimension remains constant at 256 thereafter.

The potential hidden dimensions considered for the first layer of GCN were  $d = 64, 128, 256, 384,$  and 768. Among these options, the optimal performance was observed with  $d = 256$ , therefore, subsequent experiments were conducted using this dimension.

#### Experimental settings

All experiments were carried out utilizing PyTorch<sup>47</sup>. The experimental settings are as follows: dropout rate of 0.2, learning rate of  $1e^{-6}$ , and batch size of 4. The maximum number of epochs is set to 2000. However, it is rare for training to proceed until the final epoch because we employ early stopping to mitigate overfitting. With a patience of 300, if the validation loss does not improve for 300 consecutive epochs, the training terminates, and the best model is saved as the one from 300 epochs ago.

The training time, on a PC equipped with an NVIDIA GeForce RTX 3080Ti, averages around 10 minutes per fold except embedding process. The embedding process is designed to be run only once upon receiving the dataset, enabling it to be reused later and thus excluded from the computation time. The total time for both the embedding process and similarity calculation is approximately 10 minutes for the entire ADReSSo Challenge dataset.

The experiments utilized a 5-fold cross-validation (CV) approach. During the evaluation phase with the test set, predictions from the five models chosen from each fold were combined through voting to obtain the final prediction. We assessed performance using five metrics: precision, recall, F1-score, specificity, and accuracy. Of these, accuracy was prioritized as the primary metric for performance comparison, mirroring the approach in the ADReSSo Challenge.

The ADReSSo Challenge dataset is provided with separate training and test sets, whereas the Pitt corpus is not. Previous studies targeting the Pitt corpus have employed various methods for train-test split<sup>48,49</sup>, resulting in challenges for comparing performance across studies. Therefore, the experimental results on the Pitt corpus are provided to support ablation studies rather than for direct comparison with existing models.

In experiments conducted on the Pitt corpus, the train-test split ensures that the number of AD patients and HC participants is almost equal in both the training and test sets by adjusting the random seed. The sample ratio between the training and test sets was set at 8:2.

## Results

### AD classification results on ADReSSo dataset

The performance of our model on ADReSSo Challenge test set is shown in Table 2 along with the performance of the previous SOTA architectures. Our proposed model, which utilizes only image and text modalities, achieves an accuracy of 88.73%, surpassing the previous SOTA model<sup>19</sup> that achieved 87.32% accuracy with additional features such as audio and ChatGPT's opinion. As a result of conducting a t-test for statistical significance, the

Architecture	Pr (%)	Rc (%)	F1 (%)	Sp (%)	Ac (%)
Baselines - Unimodal SOTA					
BERT-based <sup>13</sup>	81.58	88.57	84.93	80.56	84.51
BERT with handcrafted features <sup>14</sup>	–	–	–	–	84.51
Baselines - Multimodal SOTA					
BERT and acoustic models <sup>16</sup>	<b>92.00</b>	74.00	83.00	94.00	84.51
BERT with Whisper <sup>17</sup>	–	–	–	–	84.51
Co-attention with label smoothing <sup>18</sup>	84.43 ±1.59	86.29 ±4.19	85.27 ±1.78	84.43 ±2.19	85.35 ±1.44
Text, audio, and ChatGPT <sup>19</sup>	88.06	87.32	87.25	<b>94.44</b>	87.32
Proposed Graph Model					
GCN (full)	88.37 ±3.87	85.00 ±4.88	86.52 ±2.00	88.89 ±4.54	86.97 ±1.77
GCN (light) softmax	90.86 ±3.35	82.86 ±5.22	86.52 ±1.67	91.67 ±3.93	87.32 ±1.15
GCN (light)	90.93 ±2.34	85.71 ±2.33	<b>88.23 ±2.09</b>	91.67 ±2.27	<b>88.73 ±1.99</b>

**Table 2.** Performance comparison between the proposed models and state-of-the-art architectures using the ADReSSo Challenge test set. Values are presented the mean ± standard deviation. Results are averaged over four runs. *Ac* accuracy, *F1* F1-score, *Pr* precision, *Rc* recal, *Sp* specificity

Architecture	$L = 1$	$L = 2$	$L = 3$	$L = 4$	$L = 5$
Light (128)	83.10	88.73	<b>90.14</b>	83.10	83.10
Light (256)	83.10	87.32	<b>91.55</b>	90.14	76.06
Full (128)	<b>88.73</b>	84.51	<b>88.73</b>	84.51	76.06
Full (256)	<b>88.73</b>	<b>88.73</b>	81.69	81.69	83.10

**Table 3.** Accuracy for different structures with varying numbers of GCN layers. Light denotes a scenario where hidden dimensions halve repeatedly, and full denotes a scenario where hidden dimensions remain constant. In parentheses, the hidden dimension of the first layer of GCN is indicated.  $L$  represents the total number of GCN layers.

p-value was found to be  $4.3e^{-6}$ , indicating that the accuracy of our model is statistically significantly different from that of the existing SOTA model. The performance in the table represents the average of four runs, with the highest performance reaching 91.55% during one of the runs. With an F1-score of 88.23%, our model also outperforms the best F1-score of the previous SOTA model, which was 87.25%. While accuracy stands as the primary performance metric, the F1-score holds significant importance as well. Unlike specificity, the F1-score evaluates how well the model detects AD patients correctly.

The GCN with the full structure achieves an accuracy of 86.97%, which is lower than the accuracy of 88.73% attained by the light structure. This discrepancy can be attributed to the tendency of the full structure to have a larger number of parameters and a higher susceptibility to overfitting. Increasing the dropout rate to mitigate overfitting yields comparable results. Similarly, for the results of the Pitt corpus in Table 5, the accuracy of the light structure surpasses that of the full structure by 3.31% (the proposed model in the table adopts the light structure). Applying the softmax function to edge weights normalizes information propagation and emphasizes significant neighbors, enhancing training stability but not improving overall performance, as accuracy remained at 87.32%. However, it reduced fluctuations in the learning curve.

### Number of layers of GCN

Table 3 illustrates the dependency on the number of GCN layers. The variable  $L$  represents the total number of GCN layers. In parentheses, the hidden dimension of the first layer of GCN is indicated. In Table 3, the highest accuracy is generally achieved when  $L = 3$ . When  $L$  is less than 3, the model may not adequately learn the graph structure due to insufficient propagation, while for  $L$  greater than 3, oversmoothing occurs<sup>50</sup>. As the number of layers increases, instead of aggregating local information of neighboring nodes, global information of all nodes in the graph is aggregated. This results in all node embeddings on the graph becoming similar to each other, leading to oversmoothing and thereby impeding proper graph learning. Therefore, the permissible maximum number of layers decreases as the graph size decreases. In our case, the reference point is  $L = 3$ , and thus the proposed model in Table 2 is all based on  $L = 3$ .

### Removing image-text relationship

In this section, we conducted ablation studies to assess the influence of the relationship between image and text on performance by eliminating the image-text relation through three approaches: (1) shuffling edge weights, (2) independent embeddings, and (3) a combination of the first two methods.



Architecture	Pr (%)	Rc (%)	F1 (%)	Sp (%)	Ac (%)
Shuffling edge weights	81.70 ± 4.55	72.86 ± 7.56	76.87 ± 5.14	84.03 ± 4.74	78.52 ± 4.36
Independent embedding	76.52 ± 4.67	89.29 ± 4.29	82.27 ± 2.11	72.92 ± 7.31	80.99 ± 2.70
Shuffling & Ind. emb.	77.93 ± 1.66	<b>90.71 ± 2.74</b>	83.82 ± 1.71	75.00 ± 2.27	82.75 ± 1.77
Original picture	84.80 ± 3.38	75.00 ± 1.43	79.56 ± 1.16	86.81 ± 3.50	80.99 ± 1.41
Max Pooling	89.08 ± 3.92	81.43 ± 3.69	85.07 ± 3.66	90.28 ± 3.59	85.92 ± 3.45
Proposed model	90.93 ± 2.34	85.71 ± 2.33	<b>88.23 ± 2.09</b>	<b>91.67 ± 2.27</b>	<b>88.73 ± 1.99</b>

**Table 4.** Ablation study results of the ADReSSo Challenge dataset. Values presented are the mean ± standard deviation. Results are averaged over four runs. *Ac* accuracy, *F1* F1-score, *Pr* precision, *Rc* recall, *Sp* specificity

Architecture	Pr (%)	Rc (%)	F1 (%)	Sp (%)	Ac (%)
Shuffling edge weights	83.99 ± 2.37	83.61 ± 3.28	83.77 ± 2.10	80.10 ± 3.49	82.05 ± 2.27
Independent embedding	81.99 ± 3.15	<b>86.89 ± 2.68</b>	84.31 ± 1.37	76.02 ± 5.86	82.05 ± 1.87
Shuffling & Ind. emb.	81.90 ± 2.21	84.84 ± 2.06	83.30 ± 0.56	76.53 ± 3.91	81.14 ± 0.87
GCN (full)	84.45 ± 2.74	84.02 ± 2.80	84.19 ± 1.67	80.61 ± 4.25	82.50 ± 1.87
Max pooling	86.61 ± 4.10	85.66 ± 3.10	86.03 ± 1.11	<b>83.16 ± 6.74</b>	84.55 ± 1.66
<b>Proposed model</b>	<b>86.64 ± 2.59</b>	<b>86.89 ± 2.32</b>	<b>86.71 ± 0.60</b>	<b>83.16 ± 4.21</b>	<b>85.23 ± 0.87</b>

**Table 5.** Ablation study results of the Pitt corpus dataset. Values presented are the mean ± standard deviation. Results are averaged over four runs. *Ac* accuracy, *F1* F1-score, *Pr* precision, *Rc* recall, *Sp* specificity

#### Shuffling edge weights

This ablation experiment involves randomly shuffling the weights of existing edges in the bipartite graph. As meaningful connections are replaced by random ones, the image-text relation is eliminated. Experimental results demonstrate that our proposed model shows a significant improvement in accuracy by 13.00% compared to the shuffling edge weights method (see Table 4). This indicates a substantial performance enhancement, underscoring the importance of the relationship between images and texts. Similarly, experiments on the Pitt corpus show a 3.88% increase in accuracy due to proper edge weights (see Table 5).

#### Independent embeddings

When extracting embeddings using BLIP, image embeddings are influenced by text, and text embeddings are influenced by images. To mitigate this effect, there is a necessity to independently embed images and text. In the independent embedding ablation study, image embeddings were generated using the ViT<sup>51</sup>, while text embeddings were generated using Sentence Transformers<sup>52</sup>, a Python framework for sentence, text and image embeddings. The ViT models were pre-trained on the ImageNet and ImageNet-21k datasets<sup>53</sup>. Specifically, we utilized the `vit-base-patch16-224` pre-trained model to embed the image nodes, extracting the 768-dimensional embedding vector from [CLS] token of the hidden states of the last layer. For sentence nodes, we leveraged the `all-mpnet-base-v2` pre-trained model from the Sentence Transformers, which had been trained on a vast dataset comprising over one billion sentence pairs. This model was employed to generate embeddings for sentence nodes, resulting in an output of the 768-dimensional embedding vector.

The ablation results of the independent embedding are presented in Table 4 for the ADReSSo Challenge dataset and in Table 5 for the Pitt corpus. When comparing the accuracy of the proposed model with that of independent embedding, it can be observed that the embedding through VLM resulted in an improvement of 9.56% for the ADReSSo Challenge dataset and 3.88% for the Pitt corpus, respectively. While this enhancement is less pronounced than that achieved through proper edge weights, it remains a significant effect.

#### Combination of two effects

The combined impact of shuffling edge weights and independent embedding is presented in Table 4 for the ADReSSo Challenge dataset and Table 5 for the Pitt corpus. When considering the combined influence of proper edge weights and embedding through VLM, we observe an increase in accuracy of 7.23% for the ADReSSo Challenge dataset and 5.04% for the Pitt corpus. The enhancement in the ADReSSo dataset is less than when considered individually, possibly due to the random effect of shuffling, which dampens its impact. In the case of the Pitt corpus, accuracy has improved more than when considered individually, as our expectations.

#### Effect of colorization

In the case of BLIP, as the pre-training data utilized comprises the COCO dataset, it accurately provides captions for color images and demonstrates precise features along with image-text alignment. However, for grayscale drawings such as the *Cookie Theft* picture, it may provide less precise captions and may not fully exhibit proper feature extraction and image-text alignment. To evaluate the colorization effect, we conducted an ablation experiment. Table 4 presents the results, indicating a 9.56% improvement in accuracy attributable to the colorization process.

We performed colorization through minor retouching. As part of future work to enhance robustness, we intend to create several colorized images with slight variations. This ensures consistency in how participants describe the images, despite VLM perceiving them slightly differently.

### Dependence on pooling method

In our main experiment, we employed a global mean pooling to aggregate all node embeddings into a graph-level embedding. Concerned about the inclusion of unnecessary information in the averaging process, we conducted the ablation experiment using a global max pooling, which utilizes only the embeddings of the most significant nodes. Contrary to expectations, the accuracy with max pooling decreased by 3.17% compared to mean pooling, as shown in Table 4. Similarly, in the ablation experiments on the Pitt corpus presented in Table 5, replacing mean pooling with max pooling results in a decrease in accuracy of 0.80%. We attribute this decrease in performance to information loss resulting from the exclusion of less important node information when using max pooling.

## Discussion

### Critical sentences and keywords

We conducted an analysis to extract crucial sentences and keywords in classifying AD using the trained graph model. Experiments for explainability allow us to gain insights into scenarios where a participant is more likely to have AD based on specific types of sentences uttered or certain keywords frequently appearing in their speech. The method involves obtaining embedding vectors representing either the AD group or the HC group and comparing them with embedding vectors of individual sentences to investigate associations.

If we denote the representative embedding vector of the AD group and HC group as  $h_{AD}$  and  $h_{HC}$ , respectively, the process of obtaining these two vectors is as follows. Firstly, for each subject  $s$ , the pooled embedding vector  $h_s$  is computed using the best model. Then, the representative embedding vector of the AD group is calculated as  $h_{AD} = \langle h_s \rangle$  for  $s \in AD$ , and the representative embedding vector of the HC group is computed as  $h_{HC} = \langle h_s \rangle$  for  $s \in HC$ . However, in this context, subject  $s$  includes only cases where the model's prediction matches the ground truth.

#### Critical sentences for AD classification

After obtaining the representative embedding vectors for each group, comparison is conducted in two ways. The first method, similarity-based comparison, involves comparing the embedding vectors of sentences from a specific group with the representative embedding vector of the other group. For each sentence in the AD group,  $h_{s,j}^t$  for  $s \in AD$ , the cosine similarity with  $h_{AD}$  is computed. Similarly, for each sentence in the HC group,  $h_{s,j}^t$  for  $s \in HC$ , the cosine similarity with  $h_{HC}$  is computed. Extracting sentences with the highest cosine similarity values up to the top 20% yields sets of sentences  $S_{AD,\sim}$  and  $S_{HC,\sim}$ , containing sentences close to the prototype of the AD and HC groups, respectively. Thus,  $S_{AD,\sim}$  and  $S_{HC,\sim}$  represent sets of sentences crucial for distinguishing between AD and HC.

The second method, dissimilarity-based comparison, involves comparing the embedding vector of sentences from a specific group with the representative embedding vector of the other group. For each sentence in the AD group,  $h_{s,j}^t$  for  $s \in AD$ , cosine similarity with  $h_{HC}$  is computed, while for each sentence in the HC group,  $h_{s,j}^t$  for  $s \in HC$ , cosine similarity with  $h_{AD}$  is calculated. Extracting sentences with the bottom 20% lowest cosine similarity values yields sets of sentences  $S_{AD,\not\sim}$  representing AD group sentences distant from the HC group prototype and  $S_{HC,\not\sim}$  representing HC group sentences distant from the AD group prototype, which form another important set of sentences for distinguishing between AD and HC.

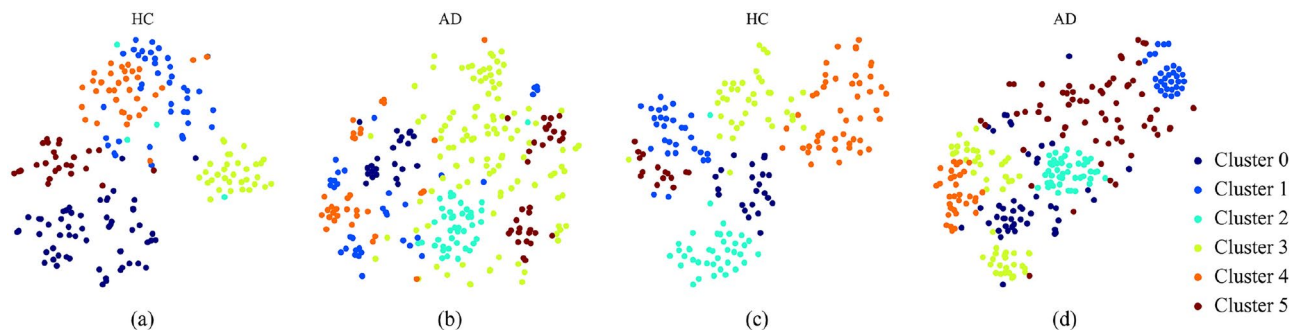
Each individual sentence in these groups is embedded using a Sentence Transformer. Subsequently,  $k$ -means clustering with six clusters is performed on each set, followed by two-dimensional visualization using t-SNE, as shown in Fig. 4. t-SNE, short for t-distributed stochastic neighbor embedding, is a statistical technique used to visualize high-dimensional data by assigning a position to each data point on a two- or three-dimensional space. Figure 4a–d represent clustering results for  $S_{HC,\sim}$ ,  $S_{AD,\sim}$ ,  $S_{HC,\not\sim}$ , and  $S_{AD,\not\sim}$ , respectively. Overall, sentences from the HC group are well-clustered, while those from the AD group exhibit a tendency to be dispersed. This is attributed to the fact that sentences from the HC group describe situations well, leading to effective clustering, whereas in the case of AD, there is a significant proportion of sentences unrelated to image descriptions, resulting in dispersion.

Examining the sentences within each cluster, shown in Fig. 5a–d, we observe common descriptions across both groups, such as *the mother is washing dishes* or *water overflowing from the sink*. However, the HC group notably contains more detailed descriptions, such as those detailing the cookie jar lid or scenes outside the window. In contrast, the AD group includes many sentences like *I don't know*. The key insight here is that these observations are facilitated by utilizing the final pooled embedding vector from the GCN.

#### Critical keywords for AD classification

Using the aforementioned approach, we analyzed the keywords essential for classifying AD and HC. From the selected sets  $S_{HC,\sim}$ ,  $S_{AD,\sim}$ ,  $S_{HC,\not\sim}$ ,  $S_{AD,\not\sim}$ , we extracted words exclusive to each group; specifically, words present only in the HC group (and vice versa for the AD group). We refer to these words as *relevant keywords*, as they play a significant role in distinguishing between AD and HC.

In contrast, while the previous steps extracted sentences up to the top 20% based on cosine similarity in the similarity-based comparison and up to the bottom 20% in the dissimilarity-based comparison, we extracted sentences up to the bottom 5% in the similarity-based comparison and up to the top 5% in the dissimilarity-based comparison. We refer to the keywords extracted from these sentences as *irrelevant keywords*, as they are not particularly helpful in distinguishing between AD and HC.



**Fig. 4.** t-SNE clusterings for (a) the HC group ( $S_{HC,\sim}$ ) and (b) the AD group ( $S_{AD,\sim}$ ) under similarity-based comparison, and for (c) the HC group ( $S_{HC,\neq}$ ) and (d) the AD group ( $S_{AD,\neq}$ ) under dissimilarity-based comparison.

#### (a) HC – Similarity-based comparison

- Cluster 0: Tell me everything that you see happening in the picture.
- Cluster 1: The water is on and the sink is spilling over.
- Cluster 2: The cookies are in the cabinet with the lid off.
- Cluster 3: The boy has taken the lid off the cookie jar, has a cookie in his left hand, is reaching with his right hand for one.
- Cluster 4: Mother is drying the dishes.
- Cluster 5: You can see a window across in the other part of the house, the roof, a tree, there are curtains on the window.

#### (b) AD – Similarity-based comparison

- Cluster 0: I don't know.
- Cluster 1: Yeah.
- Cluster 2: What else is happening here?
- Cluster 3: Oh, there's a cookie jar and a youngster with a.
- Cluster 4: Uh-huh.
- Cluster 5: The water's running over the sink.

#### (c) HC – Dissimilarity-based comparison

- Cluster 0: She kind of has her finger up towards her mouth like she might be giggling.
- Cluster 1: The sink is full of water and the water is beginning to spill over onto the floor.
- Cluster 2: I see a little boy on the stool almost falling over, taking cookies out of the cookie jar.
- Cluster 3: It looks as though the window is open and there is grass and what have you.
- Cluster 4: Okay.
- Cluster 5: The mother is drying dishes.

#### (d) AD – Dissimilarity-based comparison

- Cluster 0: Water's on the floor.
- Cluster 1: Okay.
- Cluster 2: Tell me everything that you see going on in that picture.
- Cluster 3: And I guess the mother is drying the dishes.
- Cluster 4: The kid's gonna fall off the stool.
- Cluster 5: I don't know.

**Fig. 5.** Examples of sentences corresponding to the centroids of each cluster in Fig. 4: (a)  $S_{HC,\sim}$ , (b)  $S_{AD,\sim}$ , (c)  $S_{HC,\neq}$ , and (d)  $S_{AD,\neq}$ . Sentences shaded in gray represent statements made by the investigator, those shaded in sky blue highlight characteristics of the HC group, and those shaded in pink effectively represent characteristics of the AD group.

The word cloud visualization results for these words are shown in Fig. 6. Each of Fig. 6a,b can be divided into four areas: the left represents keywords from the HC group, the right represents keywords from the AD group, the top represents relevant keywords, and the bottom represents irrelevant keywords. For instance, *cocoa* is in the AD and relevant keywords group.

Examining the trends of the keywords, notable words from the HC group, which play a significant role in distinguishing from AD, include *window*, *curtain*, *tree*, *grass*, *cabinet*, *lid*, and *counter*. On the other hand, significant words from the AD group, crucial in distinguishing from HC, include *summer*, *cocoa*, *eat*, *kid*, *lady*, and *ladder*. Words from the HC group that are not crucial in distinguishing from AD include *dish*, *woman*, *shoes*, *towel*, and *floor*, while words from the AD group that are not crucial in distinguishing from HC include *plate*, *water*, *hair*, and *hand*.

The straightforward words such as *dish*, *woman*, *water* do not play a significant role, whereas words like *curtain*, *tree*, *grass*, *lid* are crucial. This finding aligns with the research results presented in the heatmap over the area of interests on the *Cookie Theft* image<sup>42</sup>. An important takeaway from this analysis is the ability to extract words crucial for AD classification using the results of graph embeddings.

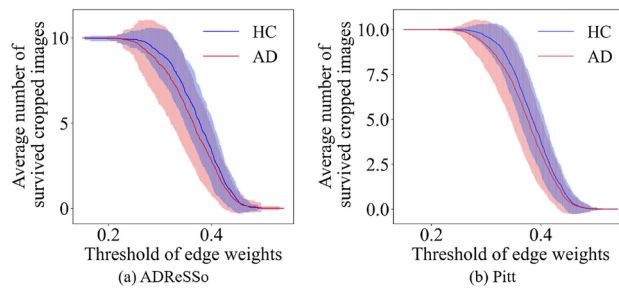
AD patients have been reported to use a reduced number of nouns and display a more limited vocabulary compared to HCs, showing an increased tendency to rely on pronouns while the diversity of nouns diminishes<sup>54</sup>, and the keywords in Fig. 6 align with this trend. For instance, AD patients often exhibit a tendency to use common nouns like 'thing,' which fail to specify concrete objects.

### Comparison quality of description in terms of graph

We conducted an analysis comparing the quality of sentences describing the picture between the AD and HC groups from a graph perspective. A sentence that effectively describes the picture would have a higher relevance with the cropped images, resulting in a higher image-text cosine similarity value, i.e., a larger edge weight. By setting a threshold and eliminating edges with weights below it, we could remove relations with low relevance



**Fig. 6.** Word clouds for (a) similarity-based comparison and (b) dissimilarity-based comparison are presented. For (a) and (b), words from the HC group (left) and the AD group (right) are shown, respectively, with performance-relevant words (top) and performance-irrelevant words (bottom).



**Fig. 7.** Threshold for edge weights versus the average number of cropped images that survive after node removal for (a) ADReSSo Challenge and (b) the Pitt corpus datasets. The HC group is presented in blue, and the AD group is presented in red. The shaded regions represent plus and minus one standard deviation.

between the image and text. During the process of edge removal, if a node with zero degree emerges, we remove that node. After measuring the remaining number of image nodes for each subject, we averaged them for each group, and the results are depicted in Fig. 7. The horizontal axis represents the threshold, while the vertical axis represents the average number of surviving image nodes after thresholding. As the threshold value increases, the average number of image nodes decreases, with the AD group showing a faster decline compared to the HC group. The Kolmogorov-Smirnov test revealed a significant difference between the AD and HC groups in the ADReSSo Challenge dataset ( $p$ -value is 0.0026). Through this analysis, we can confirm from a graph perspective that the quality of sentences describing the image is better for the HC group than for the AD group.

**Limitations**

The limitations of our proposed model lie in the manual cropping process of the picture, which may introduce subjectivity due to human intervention. Determining the optimal crop area and the ideal number of cropped images is necessary to further improve performance. By generating an optimal set of cropped images for a given picture, we can ensure high accuracy for any incoming spontaneous speech sample.

**Conclusion**

We introduce a novel approach to Alzheimer’s disease detection by leveraging both the text and image modalities of a picture description task. Our proposed method employs the VLM to construct bipartite graphs that encapsulate the relationships between image segments and corresponding textual descriptions. Our model effectively learns the structural information of the bipartite graph via the GCN. The experimental results on the ADReSSo Challenge datasets demonstrated a high accuracy of 88.73%, exceeding that of previous SOTA models. Ablation studies highlighted the critical role of the image-text relationship in enhancing classification accuracy. Additionally, the ability to identify specific sentences and keywords crucial for AD classification has significantly enhanced the explainability of our method.

For future work, we can further extend our proposed model to other types of picture description tasks, such as those found in the Delaware corpus<sup>55</sup>, a dataset used for mild cognitive impairment (MCI) screening. This dataset includes two additional pictures, *Cat Rescue* and *Going and Coming*. Incorporating additional modalities, such as audio, presents another opportunity for future research. The inclusion of embedding information from the audio modality could enhance the performance of the AD classification task.



## Data availability

The data that support the findings of this study are available from DementiaBank (<https://dementia.talkbank.org>) but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available.

Received: 12 August 2024; Accepted: 6 December 2024

Published online: 06 January 2025

## References

- Folstein, M. F., Folstein, S. E. & McHugh, P. R. “mini-mental state”: A practical method for grading the cognitive state of patients for the clinician. *J. Psychiatr. Res.* **12**, 189–198 (1975).
- Nasreddine, Z. S. et al. The montreal cognitive assessment, moca: A brief screening tool for mild cognitive impairment. *J. Am. Geriatr. Soc.* **53**, 695–699 (2005).
- Chen, S. et al. Automatic dementia screening and scoring by applying deep learning on clock-drawing tests. *Sci. Rep.* **10**, 20854 (2020).
- De la Fuente Garcia, S., Ritchie, C. W. & Luz, S. Artificial intelligence, speech, and language processing approaches to monitoring Alzheimer’s disease: A systematic review. *J. Alzheimer’s Dis.* **78**, 1547–1574 (2020).
- Vigo, I., Coelho, L. & Reis, S. Speech-and language-based classification of Alzheimer’s disease: A systematic review. *Bioengineering* **9**, 27 (2022).
- Chen, J., Ye, J., Tang, F. & Zhou, J. Automatic detection of Alzheimer’s disease using spontaneous speech only. In *Interspeech*, Vol. 2021, 3830 (NIH Public Access, 2021).
- Becker, J. T., Boiler, F., Lopez, O. L., Saxton, J. & McGonigle, K. L. The natural history of Alzheimer’s disease: Description of study cohort and accuracy of diagnosis. *Arch. Neurol.* **51**, 585–594 (1994).
- Goodglass, H., Kaplan, E. & Weintraub, S. *BDAE: The Boston Diagnostic Aphasia Examination* (Lippincott Williams & Wilkins, 2001).
- Luz, S., Haider, F., de la Fuente, S., Fromm, D. & MacWhinney, B. Alzheimer’s dementia recognition through spontaneous speech: The address challenge. *arXiv preprint[SPACE]arXiv:2004.06833* (2020).
- Luz, S., Haider, F., de la Fuente, S., Fromm, D. & MacWhinney, B. Detecting cognitive decline using speech only: The addresso challenge. *arXiv preprint[SPACE]arXiv:2104.09356* (2021).
- Balagopalan, A. & Novikova, J. Comparing acoustic-based approaches for alzheimer’s disease detection. *arXiv preprint[SPACE]arXiv:2106.01555* (2021).
- Gauder, M. L., Pepino, L. D., Ferrer, L. & Riera, P. Alzheimer disease recognition using speech-based embeddings from pre-trained models. In *Proc. Interspeech 2021* 3795–3799. <https://doi.org/10.21437/Interspeech.2021-753> (2021).
- Pan, Y. et al. Using the outputs of different automatic speech recognition paradigms for acoustic-and bert-based alzheimer’s dementia detection through spontaneous speech. In *Interspeech* 3810–3814 (2021).
- Syed, Z. S., Syed, M. S. S., Lech, M. & Pirogova, E. Tackling the addresso challenge 2021: The muet-rmit system for alzheimer’s dementia recognition from spontaneous speech. In *Interspeech* 3815–3819 (2021).
- Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint[SPACE]arXiv:1810.04805* (2018).
- Pappagari, R. et al. Automatic detection and assessment of Alzheimer disease using speech and language technologies in low-resource scenarios. *Interspeech* **2021**, 3825–3829 (2021).
- Li, J. & Zhang, W.-Q. Whisper-based transfer learning for alzheimer disease classification: Leveraging speech segments with full transcripts as prompts. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 11211–11215 (IEEE, 2024).
- Ilias, L. & Askounis, D. Context-aware attention layers coupled with optimal transport domain adaptation and multimodal fusion methods for recognizing dementia from spontaneous speech. *Knowl.-Based Syst.* **277**, 110834 (2023).
- Bang, J.-U., Han, S.-H. & Kang, B.-O. Alzheimer’s disease recognition from spontaneous speech using large language models. *ETRI Journal* (2024).
- Botelho, C. et al. Macro-descriptors for alzheimer’s disease detection using large language models. In *Interspeech 2024*, 1975–1979. <https://doi.org/10.21437/Interspeech.2024-1255> (2024).
- Radford, A. et al. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning* 28492–28518 (PMLR, 2023).
- Li, J., Li, D., Xiong, C. & Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning* 12888–12900 (PMLR, 2022).
- Baevski, A., Zhou, Y., Mohamed, A. & Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Adv. Neural Inf. Process. Syst.* **33**, 12449–12460 (2020).
- Wang, N., Cao, Y., Hao, S., Shao, Z. & Subbalakshmi, K. Modular multi-modal attention network for Alzheimer’s disease detection using patient audio and language data. In *Interspeech* 3835–3839 (2021).
- Zhu, Y., Obyat, A., Liang, X., Batsis, J. A. & Roth, R. M. Wavbert: Exploiting semantic and non-semantic speech using wav2vec and bert for dementia detection. In *Interspeech* vol. 2021, 3790 (NIH Public Access, 2021).
- Touvron, H. et al. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning* 10347–10357 (PMLR, 2021).
- Zhu, Y. et al. Evaluating picture description speech for dementia detection using image-text alignment. *arXiv preprint[SPACE]arXiv:2308.07933* (2023).
- Radford, A. et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* 8748–8763 (PMLR, 2021).
- Ilias, L., Askounis, D. & Psarras, J. A multimodal approach for dementia detection from spontaneous speech with tensor fusion layer. In *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)* 1–5 (IEEE, 2022).
- Chatzianastasis, M., Ilias, L., Askounis, D. & Vazirgiannis, M. Neural architecture search with multimodal fusion methods for diagnosing dementia. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 1–5 (IEEE, 2023).
- Rohanian, M., Hough, J. & Purver, M. Multi-modal fusion with gating using audio, lexical and disfluency features for Alzheimer’s dementia recognition from spontaneous speech. In *Interspeech 2020* 2187–2191. <https://doi.org/10.21437/Interspeech.2020-2721> (2020).
- Li, J. et al. Align before fuse: Vision and language representation learning with momentum distillation. *Adv. Neural Inf. Process. Syst.* **34**, 9694–9705 (2021).
- Li, J., Li, D., Savarese, S. & Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning* 19730–19742 (PMLR, 2023).
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **20**, 61–80 (2008).

35. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint[SPACE]* [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016).
36. Lin, T.-Y. *et al.* Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13* 740–755 (Springer, 2014).
37. Fromm, D. *et al.* The case of the cookie jar: Differences in typical language use in dementia. *J. Alzheimer's Dis.* 1–18 (2024).
38. Field, T. S., Masrani, V., Murray, G. & Carenini, G. [td-p-002]: Improving diagnostic accuracy of Alzheimer's disease from speech analysis using markers of hemispatial neglect. *Alzheimer's Dementia* 13, P157–P158 (2017).
39. Ambadi, P. S. *et al.* Spatio-semantic graphs from picture description: Applications to detection of cognitive impairment. *Front. Neurol.* 12, 795374 (2021).
40. Bouazizi, M., Zheng, C., Yang, S. & Ohtsuki, T. Dementia detection from speech: What if language models are not the answer?. *Information* 15, 2 (2023).
41. Barral, O. *et al.* Non-invasive classification of Alzheimer's disease using eye tracking and language. In *Machine Learning for Healthcare Conference* 813–841 (PMLR, 2020).
42. Mirheidari, B. *et al.* Detecting alzheimer's disease by estimating attention and elicitation path through the alignment of spoken picture descriptions with the picture prompt. *arXiv preprint[SPACE]* [arXiv:1910.00515](https://arxiv.org/abs/1910.00515) (2019).
43. Selvaraju, R. R. *et al.* Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision* 618–626 (2017).
44. Li, D. *et al.* Lavis: A one-stop library for language-vision intelligence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)* 31–41 (2023).
45. Fey, M. & Lenssen, J. E. Fast graph representation learning with pytorch geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019).
46. Morris, C. *et al.* Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 4602–4609 (2019).
47. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* 32 (2019).
48. Bertini, F., Allevi, D., Lutero, G., Calzà, L. & Montesi, D. An automatic Alzheimer's disease classifier based on spontaneous Spoken English. *Comput. Speech Lang.* 72, 101298 (2022).
49. Ortiz-Perez, D. *et al.* A deep learning-based multimodal architecture to predict signs of dementia. *Neurocomputing* 548, 126413 (2023).
50. Rusch, T. K., Bronstein, M. M. & Mishra, S. A survey on oversmoothing in graph neural networks. *arXiv preprint[SPACE]* [arXiv:2303.10993](https://arxiv.org/abs/2303.10993) (2023).
51. Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations* (2020).
52. Reimers, N. & Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint[SPACE]* [arXiv:1908.10084](https://arxiv.org/abs/1908.10084) (2019).
53. Russakovsky, O. *et al.* Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 211–252 (2015).
54. Williams, E., Theys, C. & McAuliffe, M. Lexical-semantic properties of verbs and nouns used in conversation by people with Alzheimer's disease. *PLoS ONE* 18, e0288556 (2023).
55. Lanzi, A. M. *et al.* Dementiabank: Theoretical rationale, protocol, and illustrative analyses. *Am. J. Speech-Language Pathol.* 32, 426–438 (2023).

## Acknowledgements

This research was supported by the National Research Council of Science & Technology (NST) grant by the Korean government (MSIT) (No. CAP21054-300).

## Author contributions

B.L., B.O.K and H.J.S. conceived the experiment, B.L. conducted the experiment, all authors analysed the results, B.L. wrote the manuscript. All authors reviewed the manuscript.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to B.L.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025