Multimodal Graph for Unaligned Multimodal Sequence Analysis via Graph Convolution and Graph Pooling

SIJIE MAI, School of Electronics and Information Technology, Sun Yat-sen University, China
SONGLONG XING, School of Electronics and Information Technology, Sun Yat-sen University, China
JIAXUAN HE, School of Electronics and Information Technology, Sun Yat-sen University, China
YING ZENG, School of Electronics and Information Technology, Sun Yat-sen University, China
HAIFENG HU, School of Electronics and Information Technology, Sun Yat-sen University, China

Multimodal sequence analysis aims to draw inferences from visual, language and acoustic sequences. A majority of existing works focus on the aligned fusion of three modalities to explore inter-modal interactions, which is impractical in real-world scenarios. To overcome this issue, we seek to focus on analyzing unaligned sequences which is still relatively underexplored and also more challenging. We propose Multimodal Graph, whose novelty mainly lies in transforming the sequential learning problem into graph learning problem. The graph-based structure enables parallel computation in time dimension (as opposed to RNNs) and can effectively learn longer intra- and inter-modal temporal dependency in unaligned sequences. Firstly we propose multiple ways to construct the adjacency matrix for sequence to perform sequence to graph transformation. To learn intra-modal dynamics, a graph convolution network is employed for each modality based on the defined adjacency matrix. To learn inter-modal dynamics, given that the unimodal sequences are unaligned, the commonly-considered word-level fusion does not pertain. To this end, we innovatively devise graph pooling algorithms to automatically explore the associations between various time slices from different modalities and learn high-level graph representation hierarchically. Multimodal Graph outperforms state-of-the-art models on three datasets under the same experimental setting.

CCS Concepts: • **Computing methodologies** → **Natural language generation**; • **Theory of computation** → *Machine learning theory*.

Additional Key Words and Phrases: Graph pooling, Multimodal Graph, Multimodal sequence analysis, Sentiment analysis

1 INTRODUCTION

With the development of the Internet and social platforms, there has been a large number of videos produced by users to express their views and posted online, which provides a source of multimodal data to analyze people's opinion in large quantities [3, 26, 28]. As videos are a typical form of multimodal information, people do not rely only on the spoken language, but they also use facial expressions and acoustic tones to convey information. In this paper, our downstream task is to use three modalities, i.e., language, visual and acoustic modalities, to draw inferences for the sentiment polarities of speakers [34]. These three modalities are complementary and actively interact with one another, providing more comprehensive information than one single modality.

Authors' addresses: Sijie Mai, School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China; Songlong Xing, School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China; Jiaxuan He, School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China; Ying Zeng, School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China; Haifeng Hu, School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China.

1551-6857/2022/6-ART $15.00
https://doi.org/10.1145/3542927

Hence, to maximally utilize the various information sources to capture the speaker's opinion with a multimodal architecture is a heated topic in multimodal sequence (language) analysis.

In the task of multimodal sequence analysis, two fundamental challenges exist, i.e., to learn the intra-modal dynamics of each modality, and the inter-modal counterpart for capturing cross-modal interactions [21, 59, 64]. The former relates to interactions that take place across time steps within one modality, while the latter is associated with interactions between different modalities. Previous researches mostly employ recurrent neural network (RNN) and its variants [7, 14, 19] to learn these two aspects [28, 54, 65], which, however, are slow in the inference process due to their recurrence in the time dimension. They are also prone to the problems of gradient vanishing and exploding as well as limited capacity of learning long-term dependency [4], which adds to the difficulty in learning of intra- and inter-modal dynamics. Particularly, it is of great significance to learn longer temporal dependency for unaligned sequences because they are often much longer [49].

Exploring effective approaches to learn inter-modal dynamics has been one primary focus in the research of multimodal sequence analysis. To this end, a large portion of previous works fuse the three modalities at word level [28, 32, 54, 65, 66]. However, the interactions between various modalities are usually more complicated and last for longer than one word, i.e., cross-modal interactions may take place among words and the word-level fusion may break a complete interaction into multiple parts. Additionally, word-level fusion requires that the sequences are strictly aligned at word level. However, in real-world scenarios, such word-level alignment is time-consuming and computationally expensive [63]. Therefore, we claim that a fusion strategy should be able to dynamically and automatically associate various time steps from multiple modalities instead of considering fusion at each time step.

To address the first issue, we propose to utilize the high expressiveness of graph convolution networks (GCNs) to model unimodal sequential signals as an alternative to RNN. Recently, GCNs have attracted significant attention to model graph structured data, and yielded state-of-the-art performance on a broad variety of tasks [37, 46, 56]. GCN demonstrates high effectiveness in learning the relevance of nodes via the operation of convolution, and importantly, they dispense with the need for recurrence and can be computed in parallel, which greatly boosts efficiency in inferring time compared to RNN. Existing researches on GCN mainly utilize it for modeling graph-structured data. In contrast, in this work, we extend GCN to model sequential data and comparative analysis is conducted to show that GCN exhibits greater effectiveness compared to RNN [37, 46, 56] and temporal convolutional network (TCN) variants [1, 2]. With the operation of graph convolution, longer temporal dependency can be learnt by viewing each time step as a node and associating the relevant nodes even though they are far apart in time dimension. Note that graph neural networks are also utilized in previous works to conduct fusion[31, 57, 66], they have major differences from our model (see Related Work section for more details).

However, one major obstacle to applying graph convolution to sequential learning is that sequences have no adjacency matrices as graph-structured data conventionally do. Therefore, the definition and deduction of adjacency matrices is crucial. In this paper, we design multiple ways to achieve this goal, including non-parametric methods and learnable methods. In the non-parametric way, we mainly investigate the effectiveness of a proposed matrix, namely generalized diagonal matrix, which is extremely fast and almost free of computation. In the learnable way, we automatically learn the adjacency matrix via gradient descent (direct learning) and cross-node attention (indirect learning), which is more powerful and expressive. We present the comparative results of the proposed adjacency matrices in Section 5.

For addressing the second problem, i.e., inter-modal dynamics learning, we elaborately design a graph pooling fusion network (GPFN), which learns to aggregate various nodes from different modalities by graph pooling without the need of time alignment information. Firstly we analyze the rationality of mean and max graph pooling approaches via mathematical deduction. However, mean and max graph pooling are still subject to some limitations in that they are not learnable and can only fuse neighboring nodes. Hence, to fuse the nodes in a

more expressive way, we further design a pooling strategy, termed link similarity pooling, which considers the association scores to the common neighbors of each two nodes. This learnable approach is based on the following two assumptions, i.e., (i) two nodes are closely related if their neighbors significantly overlap and thus can be fused; and (ii) provided the two nodes are neighbors, they are integrable with a high possibility. The link similarity pooling automatically updates the adjacency matrix and node embeddings for learning a high-level and refined graph representation.

In conclusion, we propose a brand-new architecture named Multimodal Graph to address the sequential learning problem. The contributions of this paper can be summarized as:

- We propose a novel graph-based architecture to model multimodal sequences, which innovatively transforms the sequential learning problem into a graph learning problem. Specifically, we design three unimodal GCNs to explore intra-modal dynamics, and a graph pooling fusion network to explore cross-modal interactions and fuse various nodes from different modalities.
- We propose multiple approaches to define adjacency matrices for sequential data, which can be extended to other sequential learning tasks. We compare the performance of different kinds of adjacency matrices empirically, and the visualization of the adjacency matrices is also provided to give insight on multimodal sequence analysis.
- In GPFN, we investigate mean/max pooling and link similarity pooling to cluster the nodes hierarchically and thus learn high-level and refined graph representation, which dispense with the need of time alignment information for modality fusion. Specially, the proposed link similarity pooling considers the common neighbors of each two nodes (the topology information in the adjacency matrix) to better fuse the nodes from different modalities. To the best of our knowledge, we are the first to leverage the power of graph pooling to conduct fusion in a hierarchical manner.
- We show that the Multimodal Graph outperforms other methods on three widely-used datasets. Besides, the contrastive experiments against RNN and TCN variants demonstrate the effectiveness of GCN on modeling sequence, which indicate a novel approach in the research of sequence modeling tasks.

2 RELATED WORK

2.1 Multimodal Sequence Analysis

Multimodal sequence analysis has attracted significant research interest in recent years [3, 65, 70]. Previous works focus on designing various fusion strategies to explore inter-modal dynamics. One of the simplest ways to explore inter-modal dynamics is to concatenate features at input feature level, which shows improvement over single modality [44, 45, 55]. In contrast, a large number of publications firstly infer decision according to each modality and combine the decisions from all modalities using some voting mechanisms [23, 39, 67]. However, as elaborated by Zadeh et al. [64], these two types of methods cannot effectively model inter-modal dynamics.

Consequently, more advanced fusion strategies are proposed in the past few years. Specifically, performing **tensor-based fusion** has received much attention [20, 27, 29, 33]. Tensor Fusion Network (TFN) [64] and Low-rank Modality Fusion (LMF) [29] adopt outer product to learn joint representation of three modalities. More recently, Mai et al. [30] propose a 'Divide, Conquer and Combine' strategy to conduct local and global fusions. **Interpretable multimodal fusion** has also received high attention recently. For example, Multimodal Routing [50] applies routing mechanism in capsule network to discover which interactions are importance for predicting each emotion, and Li et al. [26] address interpretable issue using quantum theory. Furthermore, some **translation methods** such as Multimodal Transformer (MulT) [49] aim at learning a joint representation by translating source modality into target modality. For **graph-based methods**, Graph-MFN [66] and Graph Fusion Network (GFN) [31] apply graph neural network to fuse features. Although graph neural network is used in these methods, the proposed model significantly differs from them. Firstly, they do not fuse features across the

time dimension and merely regards each modality as one node, whereas we view each time step in each modality as one node and perform graph convolution over the node embeddings. Secondly, we employ graph convolution and graph pooling for fusion as well as propose multiple adjacency matrices for sequence learning, which are not involved in Graph-MFN and GFN. More recently, Modal-Temporal Attention Graph (MTAG)[57] applies graph convolutional model to analyze multimodal sequential data, which uses dynamic pruning and read-out technique to explore cross-modal interactions and obtain multimodal embedding. However, MTAG does not comprehensively define adjacency matrix for sequences, and simply uses average operation to perform graph readout. In comparison, Multimodal Graph proposes novel graph pooling algorithm to fuse nodes and learn high-level graph representation hierarchically, and provides multiple ways to define the adjacency matrix for sequences.

To avoid sarcasm and ambiguity, many methods learn cross-modal interactions at word level such that various modalities are aligned at time dimension [6, 28, 32, 34]. For instance, Memory Fusion Network (MFN) [65] uses systems of LSTM to learn intra-modal dynamics, and it implements delta-memory attention and multi-view gated memory network to fuse memories of LSTMs across time. In addition, Multi-Fusion Residual Memory (MFRM) [32] applies multi-stage fusion to fuse the three modalities, and it designs a residual memory network to capture the long-term dependency. However, in unaligned multimodal sequence, word-level fusion cannot be performed.

A clear distinction of the majority of these previous methods between our Multimodal Graph is that we do not apply any RNN or TCN variants to learn intra-modal and inter-modal dynamics. Instead, we investigate the effectiveness of graph convolution and graph pooling on modeling multimodal sequence. Our Multimodal Graph is very elegant and effective, which can effectively learn longer temporal dependency by directly associating the distant related nodes and allow parallel computing at time dimension.

2.2 Graph Neural Networks (GNNs)

A graph can be denoted as $G = (N, E)$, where $N = \{n_1, n_2, ..., n_T\}$ is a set of nodes, and $E \subseteq N \times N$ refers to the set of observed edges. The set of node embeddings is denoted as $\boldsymbol{N} \in \mathbb{R}^{T \times d}$ where d refers to the dimensionality of each node embedding and T is the number of nodes. The edges can also be described using the adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{T \times T}$. Each element of $\boldsymbol{A}$ is non-negative and $A_{i,j} = 0$ means that nodes i and j are not connected. A wide variety of GNNs have been proposed in the last few years [11, 37, 46, 56]. We mainly focus on GCNs [25] in this paper. GCNs have become increasingly popular recently due to its applicability to graph structured data and yielded state-of-the-art performance on a variety of learning tasks, such as node classification [16], link prediction [36, 68], image retagging [48], group activity recognition [47], and graph classification [69]. GCN is basically a convolutional operation on the nodes that are directly connected. The k^{th} iteration of the general GCN can be described as:

$$(\boldsymbol{a}_i)^{k-1} = \text{AGGREGATE}((\boldsymbol{N}_j)^{k-1}; j \in \eta(i)) \tag{1}$$

$$(\boldsymbol{N}_i)^{k} = \text{COMBINE}((\boldsymbol{N}_i)^{k-1}, (\boldsymbol{a}_i)^{k-1}) \tag{2}$$

where $(\boldsymbol{N}_i)^k$ is the embedding for node i at iteration k, $\eta(i)$ represents the set of 1-hop neighbors of node i, the AGGREGATE function aggregates information from 1-hop neighbors of node i and output the aggregation representation $(\boldsymbol{a}_i)^{k-1}$, and the COMBINE function combines the information of node i and its aggregation information $(\boldsymbol{a}_i)^{k-1}$ to update the embedding of node i.

Among all GCNs, those trying to learn or recover adjacency matrices are related to our method. Franceschi et al. [10] use bi-level program to first sample adjacency matrix and then learn the parameters for the graph by minimizing inner and outer objectives. In contrast, in the direct learning way, we directly parameterize the adjacency matrix as a learnable matrix and jointly learn the adjacency matrix and graph parameters via gradient descent. Also, we provide multiple ways to define the adjacency matrix. As for graph pooling, the DiffPool [60] learns a differentiable soft cluster strategy for nodes using node embedding, mapping nodes to a set of clusters.

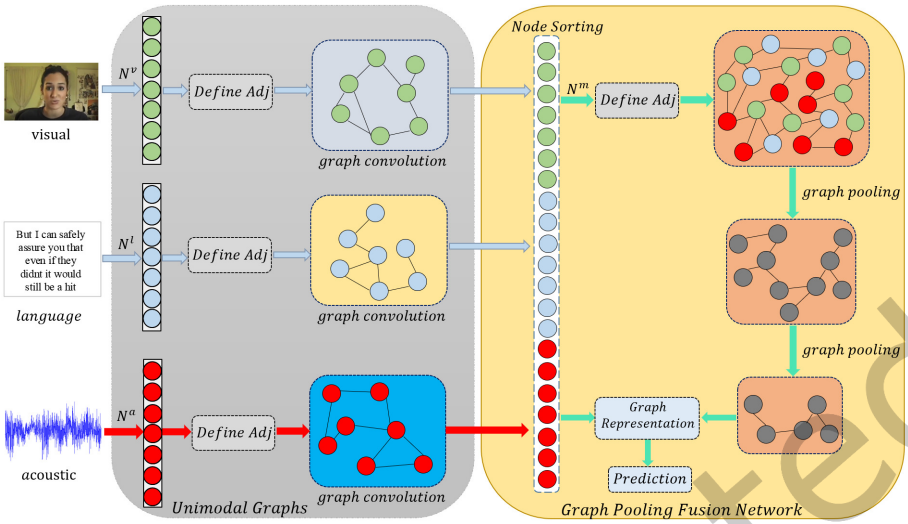


Fig. 1. **The Schematic Diagram of Multimodal Graph. Multimodal Graph consists of three Unimodal Graphs and a Graph Pooling Fusion Network (GPFN). Adj denotes adjacency matrix.**

In contrast, we utilize the adjacency matrix to learn the cluster assignment matrix, which considers the neighbor similarity of nodes. Compared to using node embedding to learn cluster assignment matrix, using adjacency matrix is more intuitive and simple. StructPool[62] uses conditional random fields to capture the high-order structural relationships among different nodes to learn a node cluster assignment matrix based on the node features, where the adjacency matrix is used to find the topological information of the graph. In contrast, we directly utilize the link (neighbor) information in the adjacency matrix to learn the cluster assignment matrix.

There also exist several works that try to apply GNNs to multimodal learning tasks. Gao et al. [12] apply fully-connected visual, semantic and numeric graphs to represent an image and conduct message passing between the graphs to utilize the contexts in various modalities. Nevertheless, the graph here is fully-connected and only represents an image. Misras et al. [38] connect every image with its corresponding tags and the image's K-nearest neighbors to capture visual-semantic relationship. Similarly, Wang et al. [53] learn a sparse multigraph construction from multimodal images. Nevertheless, these graphs are not targeted for sequential learning. Ji et al. [22] propose a cross-modal hypergraph to capture the noisy correlation among heterogeneous modalities, where the relevance of the nodes is calculated by Euclidean distance. In contrast, we propose multiple approaches to explore the relevance between nodes in the graph, and introduce graph pooling to learn refined representations for multimodal graph. Generally speaking, none of these approaches are similar to our work in terms of the main contributions: 1) we propose multiple ways to define adjacency matrix for sequences and transform the sequential learning problem into a graph learning problem; 2) we propose new graph pooling algorithms to fuse the unimodal nodes and thus learn high-level and refined graph representations; 3) we conduct extensive experiments to verify that graph-based structure can outperform popular sequential models, which indicates a new methodology in the research of sequential learning.

3 MODEL ARCHITECTURE

In this section, we describe Multimodal Graph in detail, with its diagram illustrated in Fig. 1 and the procedure is summarized in Algorithm 1. Multimodal Graph is hierarchically structured to cater to two stages, i.e., intra-modal and inter-modal dynamics learning. In the first stage, we propose to employ a graph convolution network (GCN)

Algorithm 1: Procedure for Multimodal Graph

Input: Raw sequences of unimodality $\boldsymbol{N^a} \in \mathbb{R}^{T_a \times d_a}$, $\boldsymbol{N^v} \in \mathbb{R}^{T_v \times d_v}$, and $\boldsymbol{N^l} \in \mathbb{R}^{T_l \times d_l}$.
Output: The sentiment prediction.
In Unimodal Graphs:
  For each modality do:
    Compute adjacency matrix by Eq. (3).
    Perform graph convolution by Eqs. (4)-(7).
  End
In GPFN:
  Perform node sorting to obtain multimodal sequence as in Section 3.3.1.
  Define adjacency matrix for multimodal sequence by Eq. (3).
  Perform graph convolution and graph pooling as in Fig. 2.
  Obtain graph representation as in Section 3.3.4.
  Perform prediction as in Section 3.3.4.

for each modality, termed as Unimodal Graph. In the second, a graph pooling fusion network (GPFN) is devised for capturing cross-modal interactions. As an early attempt to employ GCN for sequential learning, the architecture of Multimodal Graph is free of RNNs which are commonly used for sequential learning but subject to a number of limitations such as slowing inferring speed. Extensive discussion on applying this graph-oriented approach to sequential data is also provided in this section, including the definition of adjacency matrices. Moreover, with the proposed graph pooling algorithms in GPFN, our model can fuse the unaligned unimodal sequences and learn high-level and refined multimodal representation.

3.1 Notations and Task Definition

Table 1. Table of Notations

Notations	Descriptions
$G^a = (N^a, E^a)$	Graph for modality a with node N^a and edge E^a
$\boldsymbol{N^a}$	Sequence (node embedding) for modality a
N^m	Node set for multimodal sequence
T	The number of nodes (time steps) in a sequence
$\boldsymbol{A}$	Adjacency matrix
f	Nonlinear activation function
λ	Attenuation factor for Generalized Diagonal Matrix
ϵ	A small negative scalar to prevent division by zero
R	Matrix transposition operation
s	Pooling size for mean/max pooling

Table 2. Table of Abbreviation

Abbreviation	Full Name
GPFN	Graph Pooling Fusion Network
GCN	Graph Convolution Network
Adj	Adjacency Matrix
LSP	Link Similarity Pooling
GDM	Generalized Diagonal Matrix
TCN	Temporal Convolution Network
KNN	K-Nearest Neighbor
RNN	Recurrent Neural Network
GRU	Gated Recurrent Units
LSTM	Long Short-Term Memory Network

Our downstream task is multimodal sentiment analysis and emotion recognition. The input to the model is an utterance [40], which is a segment of a video bounded by pauses and breaths. Each utterance has three modalities, i.e., acoustic (a), visual (v), and language (l). The sequences of acoustic, visual, and language modalities are denoted as $\boldsymbol{N^a} \in \mathbb{R}^{T_a \times d_a}$, $\boldsymbol{N^v} \in \mathbb{R}^{T_v \times d_v}$, and $\boldsymbol{N^l} \in \mathbb{R}^{T_l \times d_l}$, respectively. We aim to predict the sentiment score or emotion of the utterance using the unimodal sequences. We summarize the frequently-used symbols and abbreviations in Table 1 and 2 for convenience.

3.2 Unimodal Graphs

To address the first challenge, i.e., learn intra-modal interactions, we utilize three unimodal GCNs on unimodal sequences, each for one modality. Intra-modal interactions are essential in multimodal language analysis which involves the modality-specific information. The majority of prior works leverage RNNs to explore intra-modal

dynamics [28, 32, 54, 65], which are prone to some limitations as stated in the Introduction section. Moreover, TCNs and RNNs use fixed patterns to model sequence, which are not flexible as they cannot automatically link related time steps. In contrast, we propose to use GCNs to learn unimodal high-level representations for each modality. With a suitable adjacency matrix, GCNs enable parallel computation in the time dimension and is able to learn long-term temporal contextual information by identifying and connecting related time steps.

We define the acoustic, visual, and language graphs as $G^a = (N^a, E^a)$, $G^v = (N^v, E^v)$, and $G^l = (N^l, E^l)$, respectively. Taking the acoustic graph as an example, $N^a = \{n_1^a, n_2^a, ..., n_{T_a}^a\}$ is a set of acoustic nodes, $E^a \subseteq N^a \times N^a$ refers to the set of edges that directly connects the acoustic nodes. The acoustic node embedding is denoted as $\boldsymbol{N^a} \in \mathbb{R}^{T_a \times d_a}$, which is the input acoustic sequence.

However, unlike graph-structured data, a unimodal sequence does not have an adjacency matrix that determines the graph topology. Hence, one major problem is to define the adjacency matrix in the unimodal sequence such that it effectively reflects the connection between nodes (time steps). Intuitively, two nodes are assumed to be connected if they are close in terms of the feature embedding. Therefore, we can measure the similarity between the embeddings of each two nodes to determine whether they are neighbors. Here we use a simple cross-node attention mechanism to determine the correlation between nodes and thus define the adjacency matrix for unimodal sequences. The equations are shown below (taking acoustic modality as an example):

$$\hat{\boldsymbol{A}}^{\boldsymbol{a}} = f[f(\boldsymbol{Q}\boldsymbol{W}_1^a) f((\boldsymbol{P}\boldsymbol{W}_2^a)^R)], \quad A_{i,j}^a = \frac{\hat{A}_{i,j}^a}{\sum_{v \in N} \hat{A}_{i,v}^a + \epsilon} \tag{3}$$

where $\boldsymbol{Q} = f(\boldsymbol{N^a}\boldsymbol{W}_q^a) \in \mathbb{R}^{T_a \times d}$, $\boldsymbol{P} = f(\boldsymbol{N^a}\boldsymbol{W}_p^a) \in \mathbb{R}^{T_a \times d}$, and $\boldsymbol{W}_1^a \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}_2^a \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}_q^a \in \mathbb{R}^{d_a \times d}$, $\boldsymbol{W}_p^a \in \mathbb{R}^{d_a \times d}$ are learnable matrices. f is the nonlinear activation function to increase the nonlinear expressive power of the model and R denotes the matrix transpose operation. In the learned adjacency matrix, the values can be interpreted as the intensity of interactions between nodes. Therefore, the negative links reflect little or no interaction between the corresponding two nodes. We apply *ReLU* as our activation function such that the negative links between nodes can be effectively filtered out. The equations are the same for each modality except that the node embedding to learn the adjacency matrix is different. This approach to constructing an adjacency matrix for a temporal sequence is learnable with parameters, and meanwhile it is instance-specific as it considers the various relatedness among nodes for different utterances, as opposed to directly setting all the matrix elements as learnable parameters (we will discuss it in Section 4). Hence, we term this approach indirect learning. We claim that this instance-specific and learnable approach can capture more relatedness information on the nodes and generate more favourable performance, as shown in Section 5.5. From this definition of adjacency matrix for sequential data, it can be seen that even if the two nodes are distant apart in the time dimension, they can still be directly connected if they are considered related. Therefore, compared to RNN variants, GCN can effectively learn long-term temporal dependency with fewer layers, which makes it a suitable alternative to model long temporal sequences. Compared to TCN variants that use fixed convolution pattern to model sequences, the proposed method is more flexible to automatically identify and recognize related time steps. Although the indirect learning method has some similarity to Transformer [51] in avoiding recurrence and learning long-term dependency, this approach is different from it in several aspects. Firstly, Transformer uses $softmax$ as activation function, which means that each time step is associated with all time steps. In contrast, our method is better-targeted in that it can filter out the time steps that have no direct connection, and automatically detect the one-hop neighbors for each node. Moreover, after finding the neighbors for each time step, we use a GCN such as Graph Isomorphism Network (GIN) [56] to aggregate the information of the neighbors and explore the inter-dependency between time steps, and this operation is quiet different from the feed forward network of Transformer which only operates at the feature dimension and cannot explore the connections between time steps.

Note that in common graph definition, the elements in the adjacency matrix are often binary and restricted to either 0 or 1, which denotes no/one direct connection, respectively. However, we dispense with this restriction and formulate the elements to be continuous, with a larger value indicating a closer relation between two nodes, and vice versa. This can be interpreted as multiplying an attention mask matrix to the conventional adjacency matrix. We justify the use of such continuous soft weights in adjacency matrix in Appendix.

After obtaining the adjacency matrix, the definition of COMBINE and AGGREGATE functions in GCN could have many choices. It is worth mentioning that the unimodal graphs are independent of the concrete GCN model. In other words, we can integrate any GCN model into our unimodal graph. In practice, we compare the performance of Graph Isomorphism Network (GIN) [56], Graph Attention Network (GAT) [52], GraphSAGE [16] and DiffPool [60] in our experiment. Specifically, we use GraphSAGE with mean pooling [16] as the default GCN in this paper, and the equations for the k^{th} iteration of GraphSAGE is shown below:

$$(\hat{\boldsymbol{N}}^{\boldsymbol{a}})^k = f(\boldsymbol{D}^{-1}(\boldsymbol{A}^{\boldsymbol{a}} + \boldsymbol{I})(\boldsymbol{N}_i^{\boldsymbol{a}})^{k-1}\boldsymbol{W}_a^k) \tag{4}$$

$$(\boldsymbol{N}_i^{\boldsymbol{a}})^k = \frac{(\hat{\boldsymbol{N}}_i^{\boldsymbol{a}})^k}{||(\hat{\boldsymbol{N}}_i^{\boldsymbol{a}})^k||_2} \tag{5}$$

where f is the non-linear activation function for which we use *ReLU* in our experiment, $(\boldsymbol{N}_i^{\boldsymbol{a}})^k$ is the hidden representation for node i of the acoustic modality at iteration k, and $\boldsymbol{W}_a^k$ is the parameter matrix. $\boldsymbol{D}$ is the diagonal degree matrix of $\boldsymbol{A}^{\boldsymbol{a}}$ where $D_{ii} = \sum_j A_{ij}^a$ and $D_{ij} = 0$ $(i \neq j)$. The diagonal degree matrix $\boldsymbol{D}$ is added to perform mean pooling, and the identity matrix $\boldsymbol{I}$ is added to the adjacency matrix $\boldsymbol{A}^{\boldsymbol{a}}$ to perform self-loop operation. Normalization is done in Eq. 5. To obtain the final unimodal representation, the hidden representations $(\boldsymbol{N}_i^{\boldsymbol{a}})^k$ for each layer is concatenated and sent to the fully-connected layers:

$$\boldsymbol{N}_i^{\boldsymbol{a}} \leftarrow \oplus(\boldsymbol{N}_i^{\boldsymbol{a}})^k, \ \ k \in [1, 2, .., r] \tag{6}$$

$$\boldsymbol{N}_i^{\boldsymbol{a}} = f(\boldsymbol{N}_i^{\boldsymbol{a}}\boldsymbol{W}_o^a + \boldsymbol{b}_o^a) \tag{7}$$

where $\boldsymbol{N}_i^{\boldsymbol{a}}$ is the final representation of node i for acoustic modality, $\oplus$ denotes concatenation, r is the number of layers, and $\boldsymbol{W}_o^a$ and $\boldsymbol{b}_o^a$ are the weight matrix and bias for the fully connected layers, respectively. Note that we use the same convolution structure for each modality.

With graph convolution, intra-modal interactions can be explored effectively. Unlike the commonly used RNN variants which are subject to a number of issues such as gradient vanishing and explosion, forgetting problem and slow inferring speed, GCN abandons the recurrence and can operate in parallel, which is more efficient in terms of time complexity and can learn longer temporal dependency. More importantly, it detects the immediate (one-hop) neighbors for each time step and filters out the unrelated pairs, which is better-targeted compared to Transformer. Extensive experiments are conducted in Section 5.4.1 to show the superior performance of GCN in modeling sequential data.

3.3 Graph Pooling Fusion Network (GPFN)

After exploring intra-modal interactions, the second challenge is to model inter-modal dynamics and fuse the cross-modal nodes. Considering that we focus on unaligned sequences, the common word-level fusion cannot be achieved. This means that our fusion network should learn the interactions among various nodes from multiple modalities, rather than fuse the features from three modalities at each time step. To this end, we devise GPFN to fuse the unaligned sequences. GPFN learns to aggregate the multimodal nodes to learn high-level and refined graph representations hierarchically. Specifically, in GPFN, we introduce max/mean graph pooling and analyze their rationality, which are suitable for pooling the unimodal nodes. Additionally, we propose link similarity pooling to learn a cluster assignment matrix using the link (topology) information of adjacency matrix, which can fuse the cross-modal nodes. Since features from different modalities are highly heterogeneous,

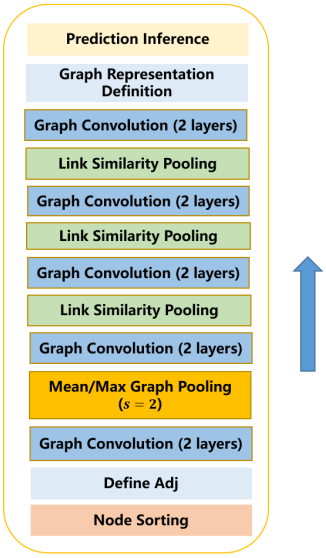


Fig. 2. **The Detailed Structure of GPFN. s denotes pooling size. The arrow in the figure indicates the direction in which the layers are stacked, from bottom to top.**

the interactions between the cross-modal nodes are much more complex than those of the nodes from a single modality. By applying link similarity pooling, the model can automatically learn the complex interactions between the heterogeneous modalities by associating the related nodes between these modalities.

As shown in Fig. 2, GPFN mainly consists of node sorting, adjacency matrix definition, graph convolution, mean/max graph pooling, link similarity pooling, graph representation definition and prediction inference. To retain consistence, the graph convolution framework and the adjacency matrix definition method are the same as those in Unimodal Graphs. In the following subsections, we will introduce node sorting, max/mean graph pooling, link similarity pooling, graph representation definition and prediction inference respectively.

3.3.1 ***Node Sorting.*** Firstly, we need to arrange the nodes to determine the order of the nodes from the three Unimodal Graphs and obtain the multimodal sequence. An intuition here is to sort nodes from three modalities according to time dimension such that the nodes from neighboring time steps are closer to each other. However, this requires that the time dimensions of different modalities are explicitly aligned. Hence, we simply concatenate the nodes at the time dimension one modality after one modality, and let the model automatically learn to aggregate these nodes. The node set N^m can be described as $N^m = \{n_1^l, n_2^l, ..., n_{T_l}^l, n_1^a, n_2^a, ..., n_{T_a}^a, n_1^v, n_2^v, ..., n_{T_v}^v\} = \{n_1^m, n_2^m, ..., n_{T_l+T_a+T_v}^m\}$. For conciseness, we denote the multimodal sequence as $\boldsymbol{N} \in \mathbb{R}^{T \times d}$ in the rest of the paper, and the adjacency matrix for multimodal sequence is denoted as $\boldsymbol{A} \in \mathbb{R}^{T \times T}$, where T is equal to $T_l + T_a + T_v$.

3.3.2 ***Mean/Max Graph Pooling.*** The initial adjacency matrix is computed in the same way as the Unimodal Graphs. With graph pooling, the related nodes are fused so that the interactions can be explored and the new adjacency matrix can be obtained. In the GPFN, we provide two pooling approaches. The first kind is the simple max pooling and mean pooling. For the adjacency matrix, 2D mean/max pooling is applied, while for the node

embeddings, 1D mean/max pooling is applied. The equations for mean pooling are given as follows:

$$\boldsymbol{N}_i^k = \frac{\sum_{g=0}^{s-1} \boldsymbol{N}_{s\cdot i-g}^{k-1}}{s} \tag{8}$$

$$A_{i,j}^k = \frac{\sum_{g=0}^{s-1}\sum_{m=0}^{s-1} A_{s\cdot i-g,s\cdot j-m}^{k-1}}{s^2} \tag{9}$$

and for max pooling:

$$\boldsymbol{N}_i^k = \max(\{\boldsymbol{N}_{s\cdot i-g}^{k-1} \mid 0 \le g \le s-1\}) \tag{10}$$

$$A_{i,j}^k = \max(\{A_{s\cdot i-g,s\cdot j-m}^{k-1} \mid 0 \le g \le s-1, 0 \le m \le s-1\}) \tag{11}$$

where s is the pooling size, $\boldsymbol{N}_i^k$ is the updated node embedding for node i at iteration k, and $\cdot$ denotes scalar multiplication. Note that in Eq. 10, the max pooling operation is element-wise.

The mean/max pooling is meaningful because the nodes in multimodal sequence are concatenated according to the time dimension of each unimodal sequence, and thus the neighboring nodes are closely related in time dimension and considered to be fusible (but we need to carefully determine the pooling size s to avoid fusing the nodes from different modalities). Moreover, we have the following observation:

Observation 1: If nodes x and y are 1-hop neighbors (directly connected), then they are 1-hop or 0-hop neighbors after mean/max graph pooling, where 0-hop neighbors mean that x and y are merged into the same node after graph pooling.

Proof: See Appendix.

The above property of mean/max pooling suggests that they are reasonable approaches for graph pooling, and indicates a principle for us to manually design graph pooling algorithms: once neighbors, always neighbors.

Although the mean/max graph pooling may be effective in fusing nodes from the same modality given that we can utilize the time information in the unimodal sequence, it is not a desirable method for fusing nodes from different modalities in unaligned multimodal sequence. Therefore, we need a learnable method that can effectively cluster the heterogeneous cross-modal nodes. To this end, we propose link similarity pooling in the following section.

3.3.3 ***Link Similarity Pooling.*** Apart from the mean/max graph pooling, we devise a learnable graph pooling method, named link similarity pooling, to leverage the link information (topology information) in the adjacency matrix to learn the node cluster assignment matrix. Different from other graph pooling methods such as DiffPool [60] that mainly utilize the node embedding to learn a cluster assignment matrix [60, 62], link similarity pooling uses the neighbor (topology) information in the adjacency matrix to learn a cluster assignment matrix, which is more intuitive and interpretable. We define the link similarity pooling in following equations:

$$\boldsymbol{Z}' = \boldsymbol{A}\boldsymbol{A}^R, \quad \boldsymbol{Z} = f[(\boldsymbol{Z}' + \boldsymbol{A})\boldsymbol{W}_z] \tag{12}$$

where R denotes matrix transposition, $\boldsymbol{W}_z \in \mathbb{R}^{T\times T'}$ is the transfer parametric matrix, and $\boldsymbol{Z} \in \mathbb{R}^{T\times T'}$ is the final node cluster assignment matrix that maps the T nodes into T' nodes (T' is sequence length after pooling). The first equation measures the similarity score of two nodes by calculating the inner product of their respective association intensity to their common neighbors. In this way, a greater extent to which a pair of nodes have similar distribution of association intensity on their shared neighbors leads to a larger score. It is also noteworthy that this score is weighted because the elements in the adjacency matrix are soft (continuous) and not restricted to be binary. For example, if nodes x and y have more common neighbors and their linked values with the common neighbor are larger, then $Z'_{x,y}$ is larger. In Fact, we have $Z'_{x,y} = \sum_{c\in CN} A_{x,c} \cdot A_{y,c}$ where CN denotes the set of common neighbors between x and y. The second equation adds the original adjacency matrix to the link

similarity information, which means that $Z_{x,y}$ is larger if x and y are 1-hop neighbors. This is reasonable because if two nodes are neighbors, then they are considered to be similar and thereby can be fused with a high possibility.

After obtaining the node cluster assignment matrix $\boldsymbol{Z}$, we use it to learn the updated adjacency matrix and node embedding. The equations are presented as follows:

$$\boldsymbol{S} = f(\boldsymbol{N}\boldsymbol{W}_s) \tag{13}$$

$$\boldsymbol{N}_{\text{update}} = \boldsymbol{Z}^R\boldsymbol{S}, \quad \boldsymbol{A}_{\text{update}} = \boldsymbol{Z}^R\boldsymbol{A}\boldsymbol{Z} \tag{14}$$

where $\boldsymbol{S}$ is the transformed node embedding, $\boldsymbol{W}_s \in \mathbb{R}^{d\times d}$ is a learnable parameter matrix, $\boldsymbol{N}_{\text{update}} \in \mathbb{R}^{T'\times d}$ and $\boldsymbol{A}_{\text{update}} \in \mathbb{R}^{T'\times T'}$ denote the updated node embedding and adjacency matrix, respectively.

The reason why we do not use the node embedding is that the adjacency matrix is originally derived from the node embedding, and therefore adjacency matrix contains part of the information in node embedding as well as the topology information. Therefore, we assume that using the adjacency matrix is sufficient to learn the node cluster matrix.

3.3.4 ***Graph Representation Definition and Prediction Inference.*** After the graph convolution and graph pooling operation of GPFN, we average the node embeddings as the representation of GPFN. Then we concatenate the representation of GPFN with the averaged node embedding from three Unimodal Graphs respectively to obtain the final graph representation. Several fully-connected layers are applied on the graph representation to infer the final sentiment decision.

4 DISCUSSION ON THE ADJACENCY MATRIX

Previously we define an indirect learning method to construct the adjacency matrix for sequential data, which is instance-specific and learnable. In this section, we aim to explore other methods to construct the adjacency matrix. We present three other ways to define the adjacency matrix, namely generalized diagonal matrix, KNN-based adjacency matrix, and a direct learning method.

4.1 Generalized Diagonal Matrix

Intuitively, the neighboring time slices are more related to each other. Therefore, we define a generalized diagonal matrix (GDM) to reflect this point:

$$\begin{pmatrix} 1 & \lambda & \lambda^2 & \dots & \lambda^{n-1} & 0 & \dots & 0 \\ \lambda & 1 & \lambda & \dots & \lambda^{n-2} & \lambda^{n-1} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \lambda^{n-1} & \dots & \lambda & 1 & \lambda \\ 0 & \dots & 0 & 0 & \dots & \lambda^2 & \lambda & 1 \end{pmatrix} \tag{15}$$

where λ is the attenuation factor which is set to 2 in our experiment, and n is the truncation factor which is set to 10. The value on the diagonal of the GDM is 1, and each element is decayed by a factor of $\frac{1}{\lambda}$ centered on the diagonal, which means the correlation between them is reduced with the increase of distance. When the distance to the diagonal element is larger than n, the value becomes zero, which means that the distant nodes no longer have direct connection. Nevertheless, we can stack many layers so that each node can still have the overall view of the input sequence. Obviously, for a sequence of length T, we need to stack $ceil((T-1)/(n-1))$ layers such that each node can incorporate information from all the nodes, where *ceil* means rounding up to an integer.

An advantage of GDM lies in that it eschews the complex computation for finding an adjacency matrix. Actually, GDM functions like the kernels in TCN [1, 2, 41]. But unlike TCN kernels, it is predefined instead of being obtained through learning (we leave the learning part to the parameters of GCN). The main differences between GCN and TCN will be discussed in Appendix. In addition, we also implement a fully-connected adjacency

matrix whose values are all ones to make a comparison. Compared to the indirect learning method, this GDM method is intuitive and fits the empirical pattern of sequence modeling, but it is neither instance-specific nor learnable. GDM serves as a reasonable baseline in the exploration of adjacency matrix for sequential data.

4.2 KNN-based method

Another simple but effective non-parametric approach to finding an adjacency matrix in a sequence is to apply K-nearest neighbor (KNN) algorithm to define the 1-hop neighbors of each node, which has also been evaluated in [10]. KNN is based on Euclidean distance, and we select the nodes with shortest distance as the 1-hop neighbors for each node. The equations can be described as below:

$$d_{ij} = \frac{1}{\mathrm{Eur}(\boldsymbol{N}_j; \boldsymbol{N}_i) + \epsilon} \tag{16}$$

$$\hat{A}_{i,j} = \mathrm{ReLU}(d_{ij} - \alpha \times \frac{\sum_j d_{ij}}{T}) \tag{17}$$

$$A_{i,j} = \frac{\hat{A}_{i,j}}{\sum_{v \in N} \hat{A}_{i,v}} \tag{18}$$

where Eur denotes the Euclidean distance, d_{ij} denotes the 'similarity' of node j to node i, ϵ denotes a positive scalar to prevent division by zero, and α is a scalar that controls to what extent the weak links can be filtered out. Eq. 17 filters out the links that have no strong connection and Eq. 18 denotes simple normalization. Note that by applying Eq. 17, we do not select an exact number of k neighbors for each node, but we allow a variable number of neighbors, as long as the links between the node and its neighbors meet the threshold. A disadvantage of KNN-based method is that it has to compute the Euclidean distance of each two nodes for each instance, making it more time-consuming compared to GDM. By definition, KNN-based adjacency matrix is instance-specific, but it is not learnable, serving as a reasonable comparative method to the indirect learning method.

4.3 Direct Learning

Another intuitive way to construct an adjacency matrix is to learn the matrix directly in the training process. In this method, the adjacency matrix is parameterized as a learnable matrix $\hat{\boldsymbol{A}} \in \mathbb{R}^{T \times T}$, and a ReLU function is applied to activate the learned matrix: $\boldsymbol{A} = \mathrm{ReLU}(\hat{\boldsymbol{A}})$ such that the linked values between nodes are non-negative.

To obtain a more representative adjacency matrix, we add a regularization term, as shown below:

$$\ell = \sum_{i \in N} (\sum_{j \in N} \hat{A}_{i,j})^2 + \sum_{i \in N} (\sum_{j \in N} \mathrm{ReLU}(\hat{A}_{i,j}) - \gamma)^2 \tag{19}$$

where the first term forces the sum of linked values of each node to equal zero, which can prevent the learned matrix from becoming a fully-connected matrix; the second term restricts the sum of the positive linked values of each node to approximate a given positive scalar γ so as to prevent it from degenerating into an all-zero matrix. ℓ is optimized via gradient descent. Note that the learnt adjacency matrix is shared across instances (instance-independent), which may not be expressive enough compared to the instance-specific indirect learning method but is computationally efficient.

5 EXPERIMENTS

Multimodal Graph is evaluated on three popular datasets for multimodal learning. In this section, we focus on the following questions: 1) Does Multimodal Graph perform favorably to TCN and RNN variants? 2) Does Multimodal Graph achieve state-of-the-art performance on multimodal sentiment analysis and emotion recognition? 3) What kind of GNNs performs best in our Multimodal Graph? 4) What kind of adjacent matrix performs best? 5) What are the attributes of adjacent matrices?

5.1 Datasets

5.1.1 ***CMU-MOSI***. CMU-MOSI [67] is a widely-used dataset for multimodal sentiment analysis. It contains 93 videos in total, and each video is divided into 62 utterances at most. The intensity of sentiment ranges within [-3,3], where -3 indicates the strongest negative sentiment, and +3 the strongest positive. We evaluate model's performance using various metrics, including 7-class accuracy (i.e., Acc7: sentiment score classification), binary accuracy (i.e., Acc2: positive or negative sentiments), F1 score, mean absolute error (MAE) of the score, and the correlation of the model's prediction with humans (Corr). To be consistent with prior works, we use 1,284 utterances for training, 229 for validation, and 686 for testing.

5.1.2 ***CMU-MOSEI***. CMU-MOSEI [66] is the largest multimodal language analysis dataset that contains a total number of 2928 videos. The dataset has been segmented at the utterance level, and each utterance has been scored on two levels: sentiment ranging between [-3, 3], and emotion with six different values. We use the sentiment label in our task. In our experiment, the evaluated metrics for CMU-MOSEI are the same as those for CMU-MOSI dataset. We use 16,265 utterances as training set, 1,869 utterances as validation set, and 4,643 utterances as testing set.

5.1.3 ***IEMOCAP***. IEMOCAP [5] is a multimodal emotion recognition dataset that contains a total number of 151 videos from 10 speakers. The videos are segmented into about 10K utterances. IEMOCAP has the following labels: anger, happiness, sadness, neutral, excitement, frustration, fear, surprise and other. We take the first four emotions to compare with our baselines. We follow previous works[49, 65] to report the classification accuracy and the F1 score of each emotion.

5.2 Baselines

The baselines for comparison include **Early Fusion LSTM** (**EF-LSTM**), **Late Fusion LSTM** (**LF-LSTM**), **Tensor Fusion Network** (**TFN**) [64], **Memory Fusion Network** (**MFN**) [65], **Multi-Fusion Residual Memory Network** (**MFRM**) [32], **Multimodal Transformer** (**MulT**) [49], and **Modal-Temporal Attention Graph** (**MTAG**) [57]. Notably, for EF-LSTM, MFN, and MFRM, since they adopt word-level fusion, connectionist temporal classification (CTC) [15] is performed to process the unaligned sequences to obtain approximately aligned sequences. The models are trained to optimize the CTC alignment objective and the prediction objective simultaneously. The detailed introduction of baselines are shown in Appendix for lack of space.

5.3 Experimental details

Hyperparameter Setting: We develop our model on Pytorch with RTX2080Ti as GPU. We apply Mean Absolute Error (MAE) as loss function with Adam [24] as optimizer (the loss function for IEMOCAP is cross-entropy loss). The defaulted adjacency matrix is the indirect learning one. For the hyper-parameter setting, please refer to Table 3.

Baseline Evaluation: To ensure a fair comparison, for each baseline, following Gkoumas et al.[13], we reproduce the codes and determine the hyperparameters of the baseline by performing fifty-times random grid search on the hyperparameters, and the hyperparameter setting that reaches the best performance is saved. After the hyperparameters are determined, we train the model again with the best hyperparameters for five times with different random seeds, and the final results are obtained by calculating the mean results of the five-time running. Our method follows the same procedure to obtain the final results. **Feature Extraction:** For feature extraction, to make a fair comparison with baselines, we follow the setting of CMU-MultimodalSDK [1]. GloVe word embeddings [42] are used to extract the features of the transcripts in the videos. The Glove word embeddings, represent each word as a 300-dimensional vector, are trained on 840 billion tokens from the common crawl

[1] https://github.com/A2Zadeh/CMU-MultimodalSDK

Table 3. Hyperparameters of Multimodal Graph.

	CMU-MOSI	CMU-MOSEI	IEMOCAP
Batch Size	50	50	24
Initial Learning Rate	0.001	0.001	0.001
Training Epochs	50	20	15
Gradient Clip	0.8	0.8	0.8
Pooling Sequential Length (T')	70	75	50
Feature Dimensionality (d)	50	40	80
Convolution Layers (r)	2	2	2

Table 4. **Comparison with RNN and TCN variants.** The GRU and LSTM models used here are bidirectional. Training Time means training time of the model per batch (the batch size is the same for all models).

	Acc2	Acc7	F1	MAE	Corr	Training Time (s)	Parameters
GRU	80.4	**50.5**	80.5	0.611	0.669	0.933	1,018,811
LSTM	80.7	48.3	81.3	0.623	0.662	0.774	**917,831**
Regular TCN	62.9	41.4	77.2	0.838	0.007	**0.120**	1,307,071
1D-ResNet	79.8	48.4	80.1	0.626	0.646	0.134	1,367,561
Dilated 1D-ResNet	80.4	48.0	80.7	0.636	0.644	0.127	1,367,561
Multimodal Graph	**81.4**	49.7	**81.7**	**0.608**	**0.675**	0.125	1,225,400

dataset. Facet[2] is used to extract a sequence of visual features that are composed of facial action units, facial landmarks, head pose, and gaze tracking, etc. COVAREP [8] is utilized for extracting acoustic features including 12 Mel-frequency cepstral coefficients, pitch tracking, speech polarity, glottal closure instants, spectral envelope, etc. These acoustic features are extracted from the full audio clip of each utterance to form a sequence that represents variations in the tone of voice across the utterance. We refer the reader to [49] for more details about the features.

5.4 Comparative Results

5.4.1 ***Comparison with TCN and RNN.*** The mainstream approaches to processing sequences are RNN and TCN variants. Here we compare the proposed Multimodal Graph with RNN and TCN variants where the Unimodal Graphs and GPFN are replaced by them so as to investigate the effectiveness of GNNs on modeling sequence (for Transformer [51], please refer to Section **??** to see the comparison with Multimodal Transformer [49]). The TCN variants for comparison include the regular TCN that applies several 1-dimensional convolution layers to process the sequences[2], the ResNet counterpart that applies 1-dimensional convolution (1D-ResNet)[18], and the ResNet counterpart that applies 1-dimensional dilated convolution[61] (Dilated 1D-ResNet). The RNN variants for comparison include GRU [7] and LSTM [19].

We can infer from Table 4 that GRU [7] and LSTM [19] perform competitively, and Multimodal Graph still outperforms GRU and LSTM across the majority of the evaluation metrics. Specifically, Multimodal Graph reaches the best performance on binary accuracy, F1 score, MAE, and Corr metrics, and ranks second on the 7-class accuracy. For the TCN variants, the regular TCN without residual learning obtains the worst performance. 1D-ResNet and Dilated 1D-ResNet obtain satisfactory results and outperform the regular TCN by a significant margin, demonstrating the effectiveness of residual learning on building up a deep convolutional network to model sequences. Nevertheless, our Multimodal Graph still outperforms 1D-ResNet and Dilated 1D-ResNet,

[2] iMotions 2017. https://imotions.com/

Table 5. **Performance of Multimodal Graph on CMU-MOSI and CMU-MOSEI datasets.**

	CMU-MOSI					CMU-MOSEI				
Methods	Acc2	Acc7	F1	MAE	Corr	Acc2	Acc7	F1	MAE	Corr
LF-LSTM	74.5	31.3	74.3	1.042	0.608	79.5	48.0	79.6	0.632	0.650
EF-LSTM	73.7	32.2	73.5	1.038	0.594	65.3	41.7	76.0	0.799	0.265
TFN [64]	77.9	32.4	75.0	1.040	0.616	79.5	49.3	78.9	0.613	0.673
MFN [65]	77.7	30.9	75.5	1.032	0.627	80.6	49.1	80.0	0.612	**0.687**
MFRM [32]	79.8	34.7	79.4	0.956	0.673	80.3	48.9	80.6	0.617	0.669
MulT [49]	**80.6**	**35.3**	79.3	0.972	0.681	80.1	49.0	80.9	0.630	0.664
MTAG [57]	80.5	31.9	80.4	0.941	**0.692**	79.1	48.2	75.9	0.645	0.614
Multimodal Graph	**80.6**	32.1	**80.5**	**0.933**	0.684	**81.4**	**49.7**	**81.7**	**0.608**	0.675

Table 6. **Comparison between Multimodal Graph and other approaches on IEMOCAP dataset.**

	Happy		Sad		Angry		Neutral		Average	
Models	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
MFN [65]	79.7	78.9	73.1	71.7	72.8	66.9	57.9	56.5	70.9	68.5
TFN [67]	77.1	76.8	68.9	63.2	71.7	66.8	51.0	51.2	67.2	64.5
MulT [49]	85.7	79.4	79.3	70.5	75.8	65.4	51.8	52.2	73.2	66.9
MFRM [32]	85.9	79.6	76.7	71.4	76.0	68.9	55.1	53.2	73.4	68.3
MTAG [57]	85.8	79.7	**79.5**	**72.8**	76.2	69.8	57.8	56.9	74.8	69.8
Multimodal Graph	**86.0**	**81.3**	79.4	72.3	**76.8**	**71.9**	**61.4**	**59.8**	**75.9**	**71.3**

yielding over 1% improvements on Acc7 and Acc2. These results demonstrate Multimodal Graph's superiority in modeling sequences. This is partly because Multimodal Graph can automatically learn long-term dependency with a suitable adjacency matrix that links distant related time steps. Moreover, different from the fixed modeling patterns of the TCNs and RNNs, for each time step, Multimodal Graph can identify and link various related time steps with it, which is more interpretable, flexible, and representative (see Appendix for detail). These results demonstrate GCN with an appropriate adjacency matrix as a novel and effective way of modeling sequences.

Analysis of Training Time and Parameters: Intuitively, since graph convolution dispenses with the recurrence nature and allows parallel operation in the time dimension, it is faster than RNN networks. To verify this point, we report the training time per batch on the CMU-MOSEI dataset, as shown in Table 4. It can be seen that under the same experimental setting, training Multimodal Graph requires only 0.125s per batch, compared to 0.933s and 0.774s per batch for GRU and LSTM, respectively. Moreover, the training time of the three convolution networks (regular TCN, 1D-ResNet and Dilated 1D-ResNet) is close to that of our model. This empirically demonstrates that GCN is much more efficient than RNNs and is comparable to TCNs in terms of time complexity without sacrificing performance. We also report the number of trainable parameters to evaluate the space complexity of the models. It can be seen that GRU and LSTM require 1,018,811 and 917,831 parameters respectively, both fewer than GCNs. This is because we only implement three layers of GRU/LSTM which performs best according to our experiment. When we stack more layers, the performance of GRU/LSTM decreases, which is reasonable because RNNs generally are more difficult to train when model grows deeper. Additionally, the number of parameters of the three TCNs is slightly larger than that of our Multimodal Graph, demonstrating that the improvement of our model is not simply due to the increase in the number of parameters.

5.4.2 ***Comparison with Baselines for Multimodal Sentiment Analysis***. We compare our Multimodal Graph with the competitive baselines on two benchmark datasets CMU-MOSI [67] and CMU-MOSEI [66],

and the results are presented in Table 5. It can be seen that Multimodal Graph yields best results on most of the metrics on two datasets. Specifically, our model surpasses the state-of-the-art unaligned fusion method MulT[49] in terms of all metrics, except 7-class accuracy on CMU-MOSI. The results demonstrate the effectiveness of graph convolution and graph pooling in learning sequential data, compared to Transformer[51] which is employed in MulT. Moreover, our model also outperforms the graph-based method MTAG[57] across the majority of the evaluation metrics, further demonstrating the superiority of our method. We argue that this is partly because compared to MTAG that averages the nodes in the graph to perform graph readout (fusion), our Multimodal Graph applies graph pooling and devises novel pooling method to aggregate the multimodal nodes and learn high-level graph representation hierarchically.

Notably, the recent state-of-the-art aligned models [17, 35, 58] make great progress by using the large-pretrained BERT [9] to extract the language representation. In contrast, to make a fair comparison, we follow the state-of-the-art unaligned fusion models to use GloVe [42] to extract language embedding. Our method can also be extended to the aligned setting and uses BERT [9] to reach remarkable results. In our experiment, our model achieves a binary accuracy of 85.9% and a 7-way accuracy of 48.7% on CMU-MOSI dataset under the aligned setting, which suppresses the state-of-the-art models [17, 58]. Since our focus is unaligned fusion in this paper, we do not present the detailed results of aligned setting due to the lack of space.

Additionally, we discover that our Multimodal Graph performs better on larger dataset (CMU-MOSEI) than smaller one (CMU-MOSI). Similarly, the model performs better on IEMOCAP (see Section 5.4.3), which is a larger dataset, than on CMU-MOSI. These results show good generalizability of our model, which is scalable to large datasets.

5.4.3 ***Comparison with Baselines for Multimodal Emotion Recognition***. We additionally evaluate the proposed method on the task of multimodal emotion recognition to justify the generalization ability of the model to other multimodal learning task. The widely-used dataset IEMOCAP is evaluated in this section. From the results presented in Table 6, it can be seen that the Multimodal Graph outperforms the baselines in the tasks of recognizing the 'Happy', 'Angry' and 'Neutral' emotions, yielding about 3.5% improvements compared with the best results of baselines on the recognizing of the 'Neutral' emotion. More importantly, Multimodal Graph outperforms the baselines in terms of the average performance, yielding over 1% improvements on the average accuracy and average F1 score. In addition to the task of multimodal sentiment analysis, the extra experiments of the more challenging multimodal emotion recognition task have proven the effectiveness and generalization ability of the proposed Multimodal Graph.

5.4.4 ***Ablation Studies***. In this section, we perform ablation studies to investigate the effectiveness of the proposed graph convolution and graph pooling. From the results in Table 7, we discover that the introduced graph convolution and graph pooling are both beneficial to the performance of the model, without which the performance drops considerably. The graph convolution brings greater improvement than the graph pooling, indicating the importance of learning more expressive and discriminative unimodal representations. We also investigate the influence of the proposed link similarity pooling by removing it from the GPFN (see the case of 'W/O LSP in GPFN' in Table 7). The results suggest that the link similarity pooling is effective, which demonstrates the importance of a learnable graph pooling algorithm to automatically identify and cluster the related nodes in the multimodal graph hierarchically.

5.4.5 ***Analysis of Model Complexity***. To analyze the model complexity of Multimodal Graph, we use the number of trainable parameters as the proxy for its space complexity, and compare it with the state-of-the-art unaligned fusion methods, as reported in Table 8. It can be seen that our Multimodal Graph requires 1,225,400 trainable parameters on CMU-MOSEI, which is 64.46% of the number of parameters of MulT. Although Multimodal Graph requires fewer trainable parameters than the current state-of-the-art model MulT, it still outperforms MulT

Table 7. **Ablation studies on CMU-MOSEI.** In the case of 'W/O Graph Pooling', we remove the graph pooling layers in the GPFN. In the case of 'W/O Graph Convolution', we replace the graph convolution network with the fully connected layer to map the feature dimensionality of different modalities to be the same. 'W/O LSP in GPFN' denotes that the link similarity pooling is removed from the GPFN.

	Acc2	Acc7	F1	MAE	Corr
W/O Graph Convolution	79.7	48.2	80.1	0.639	0.645
W/O Graph Pooling	80.4	49.6	80.5	0.612	0.671
W/O LSP in GPFN	80.8	49.2	81.1	0.610	0.674
Multimodal Graph	**81.4**	**49.7**	**81.7**	**0.608**	**0.675**

Table 8. **The Comparison of Model Complexity on CMU-MOSEI.** We implement three widely-used baselines for comparison in this section.

Methods	TFN [64]	MulT [49]	MFN [65]	Multimodal Graph
Number of Parameters	1,002,471	1,901,161	**963,777**	1,225,400

Table 9. **Discussion on the Concrete Graph Neural Networks on CMU-MOSEI.** For the case of 'GAT' and 'GIN', we replace the defaulted graph convolution model GraphSAGE in Unimodal Graphs and GPFN with the corresponding GAT and GIN model. For the case of 'DiffPool', we replace the max/mean graph pooling and link similarity pooling in GPFN with DiffPool.

Models	Acc2	Acc7	F1	MAE	Corr
GAT [52]	80.3	49.0	80.4	0.627	0.650
GIN [56]	81.1	49.0	81.6	0.615	**0.676**
GraphSAGE [16]	**81.4**	**49.7**	**81.7**	**0.608**	0.675
DiffPool [60]	81.1	**49.8**	81.3	0.611	0.670
GPFN	**81.4**	49.7	**81.7**	**0.608**	**0.675**

as shown in Section 5.4.2. This advantage in space complexity is significant because it shows that in addition to being effective in modeling sequential data, our Multimodal Graph is also more efficient than the variant of Transformer which is one dominant sequence learning method. This further validates GCN as a novel way of modeling sequential data. Compared to TFN and MFN, the proposed Multimodal Graph has more parameters. To sum up, given the high empirical performance of Multimodal Graph, the space complexity of Multimodal Graph is moderate compared to state-of-the-art unaligned sequence analysis methods.

5.4.6 ***Discussion of Different Graph Neural Networks.*** Since our Multimodal Graph is independent of the concrete GCN structure, we can choose any GCN structure to implement our model. This subsection compares different current GCN structures to analyze which kind of GCN structures performs best. Specifically, we compare the defaulted GraphSAGE [16] with Graph Attention Network (GAT) [52] and Graph Isomorphism Network (GIN) [56]. From Table 9 it can be seen that GraphSAGE [16] reaches the best performance among all the compared GCNs. In addition, the results of GIN [56] and GAT [52] are also satisfactory, indicating the generalization ability of Multimodal Graph.

As for the graph pooling methods, we implement and evaluate DiffPool [60] as the baseline to compare with the proposed GPFN. DiffPool [60] achieves a relatively good result and is still inferior to our model. To be more

Table 10. **Discussion on the adjacency matrices on CMU-MOSEI.**

	Acc2	Acc7	F1	MAE	Corr
All-one Matrix	78.8	49.0	79.3	0.644	0.623
KNN	80.3	45.8	80.6	0.659	0.625
GDM	81.1	49.3	81.2	0.617	0.666
Direct Learning	80.7	49.3	81.2	0.618	0.659
Indirect Learning	**81.4**	**49.7**	**81.7**	**0.608**	**0.675**

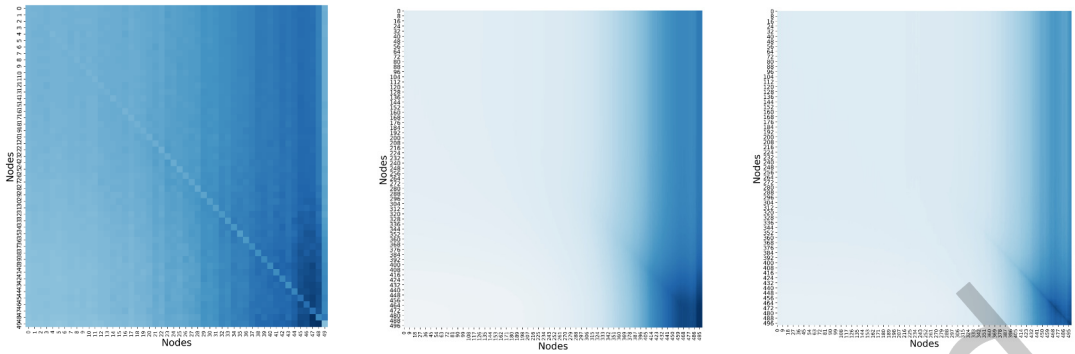
specific, DiffPool outperforms our GPFN in terms of 7-class accuracy by 0.1 points, while our model outperforms DiffPool on the rest of the evaluation metrics. These results demonstrate the effectiveness of our graph pooling method.

5.5 Discussion of Adjacency Matrices

5.5.1 ***The Comparison of Different Adjacency Matrices.*** To analyze the performance of different kinds of adjacency matrices, we conduct an experiment where different types of adjacency matrix are used in Multimodal Graph. To show the effectiveness of the proposed types of adjacency matrix, i.e., indirect learning, direct learning, GDM and KNN-based adjacency matrices, we additionally implement one comparative trial where the adjacency matrix is an all-one matrix which corresponds to a fully-connected graph. We can infer from Table 10 that among all the adjacency matrices, the all-one matrix performs worst, which is understandable because the GCN in this case will reduce into a fully-connected graph with no ability to discern various relatedness between nodes. Besides, the KNN-based adjacency matrix reaches a relatively low performance, indicating that the Euclidean distance is not a perfect choice to determine the correlation between heterogeneous nodes. In contrast, both the indirect learning method and the GDM achieve satisfactory results. This is because GDM is in line with the general pattern of sequence modeling which sticks out the present time step and dilutes the past ones. And the direct learning method can learn such pattern in the absence of prior knowledge by gradient descent. Additionally, the indirect learning method, produces best results on all metrics. One possible reason is that it is both learnable and instance-specific, and therefore it is more representative and can be optimized to capture more subtle and complex relatedness among nodes. To sum up, the comparative results that different types of adjacency matrices yield suggest that: (1) the proposed adjacency matrices are helpful in capturing useful information on the relatedness among nodes, compared to an all-one adjacency matrix that contains no such information, and (2) a learnable and meanwhile instance-specific adjacency matrix is crucial for capturing more relatedness information among nodes.

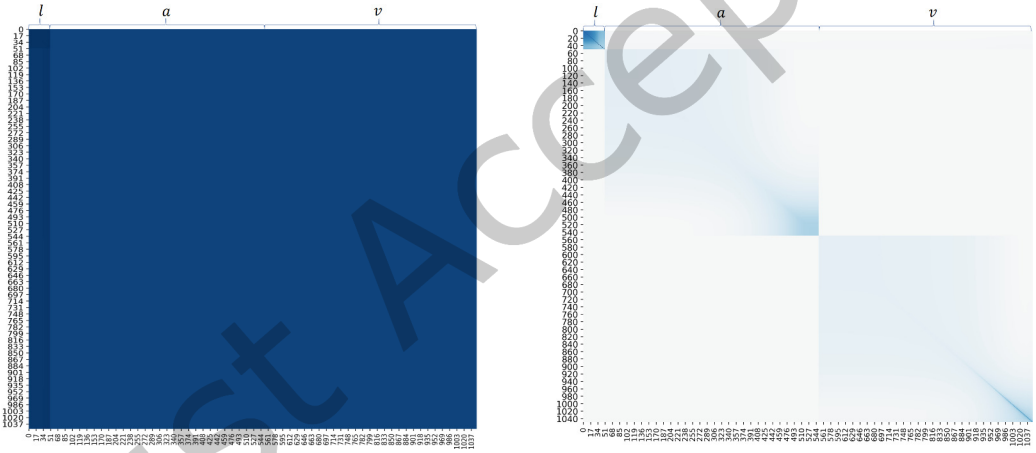
5.5.2 ***Visualization of Adjacency Matrices.*** To analyze the attributes of the indirect learning adjacency matrices, we provide a visualization of the unimodal and multimodal indirect learning adjacency matrices. Instead of merely analyzing a few instances, we average the adjacency matrices of all the testing instances in CMU-MOSEI to reveal the general patterns of the adjacency matrices, which is more convincing and informative.

Unimodal Adjacency Matrices: As can be inferred from Fig. 3, for the visual and acoustic modalities, the latter portion of nodes have much more impact than their counterparts in that they have more intensive link association (the nodes tends to have more connections with the latter portion of nodes), indicting that the model heavily relies on the latter portion of nodes for prediction and they are more informative. In contrast, the link association between different nodes in language modality is more even, indicting that the connections between different words are distributed evenly at nearly all temporal positions. Interestingly, the language adjacency



(a) Indirect Learning (Language) (b) Indirect Learning (Acoustic) (c) Indirect Learning (Visual)

Fig. 3. **Visualization of Unimodal Indirect Learning Adjacency Matrices.** Each grid in the figure reflects the interaction between each corresponding two nodes. A darker color reflect a stronger interaction between the corresponding two nodes, and vice versa. For indirect learning, since the adjacency matrices are instance-specific, we average the adjacency matrices of the testing instances and visualize the mean adjacency matrix.



(a) Indirect Learning (Cross-modal) (b) KNN (Cross-modal)

Fig. 4. **Visualization of Cross-modal Indirect Learning Adjacency Matrices.** A darker color indicates a stronger interaction, and vice versa. The labels of x and y axes are both 'nodes'. Since the KNN-based and indirect learning methods are both instance-specific, we average the adjacency matrices of the testing instances and visualize the mean adjacency matrices. The first 50, the middle 500 and the latter 500 nodes belong to language, acoustic, and visual modality, respectively.

matrix suggests that the language nodes have fewer self-connections. This is reasonable because during the graph convolution, we add self-loop operation such that the nodes can connect with themselves (see Eq. 4), so the adjacency matrix does not need to learn a strong self-connection value on the diagonal, otherwise it would be redundant.

Cross-modal Adjacency Matrices: We also conduct visualization of the indirect learning adjacency matrix for multimodal sequence, and we provide the visualization of KNN-based adjacency matrix for comparison. As

can be inferred from Fig. 4, for KNN-based method, the nodes from different modalities have no direct or close interactions with each other, and these nodes only interact with their neighboring nodes that come from the same modality. This means that by using the KNN-based adjacency matrix, the cross-modal interactions cannot be effectively explored, which may explain why it performs worse than other methods. The visualization of the KNN-based adjacency matrix also suggests that the distribution gap between different modalities actually exists and we need to handle it during modality fusion [31]. In contrast, for adjacency matrix of the indirect learning method, interestingly, the mean adjacency matrix for multimodal sequence suggests that the nodes tend to connect with the language nodes (i.e., the first 50 nodes) more closely, which indicates that the language nodes are more important and informative. This scenario can be partly explained by the fact that language is more important than the other modalities, as revealed in [30, 43]. Additionally, as the adjacency matrix for the language nodes (the first 50 nodes) has the darkest color, our visualization suggests the language nodes have the strongest connection with each other in the multimodal sequence.

6 CONCLUSION

We employed popular GNNs to process multimodal sequences, which was free of the recurrent structure and proved more efficient and competitive. Specifically, we developed a unimodal graph for each modality to explore intra-modal dynamics, and a graph pooling fusion network over Unimodal Graphs to learn inter-modal dynamics. We proposed multiple ways to construct an adjacency matrix for a sequence. The experiments suggested that: 1) the proposed GCN-based model outperformed RNN and TCN variants with high computational efficiency, which indicated a new research direction in modeling sequences; 2) the proposed learnable and instance-specific methods to define an adjacency matrix performed best; 3) our method outperformed the transformer-based model MulT, which indicated the effectiveness of graph-based models; 4) the visualization results suggested that our model can identify important intra- and inter-modal interactions; 5) the proposed GPFN outperformed the classical graph pooling method DiffPool in fusing cross-modal information. However, Multimodal Graph is non-causal, and thus not applicable to some language processing tasks. In the future, we aim to develop a causal GCN-based model to process sequences.

7 ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 62076262.

REFERENCES

[1] Shaojie Bai, J. Kolter, and Vladlen Koltun. 2019. Trellis Networks for Sequence Modeling. In *Proceedings of International Conference on Learning Representations*.

[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *Arxiv preprint Arxiv: 1803.01271* (2018).

[3] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2019. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 2 (Feb 2019), 423–443. https://doi.org/10.1109/TPAMI.2018.2798607

[4] Y Bengio, P Simard, and P Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5, 2 (1994), 157–166.

[5] Carlos Busso, Murtaza Bulut, Chi Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. 2008. IEMOCAP: interactive emotional dyadic motion capture database. *Language Resources and Evaluation* 42, 4 (2008), 335–359.

[6] Minghai Chen, Sen Wang, Paul Pu Liang, Tadas Baltrus̆aitis, Amir Zadeh, and Louis Philippe Morency. 2017. Multimodal sentiment analysis with word-level fusion and reinforcement learning. In *19th ACM International Conference on Multimodal Interaction (ICMI'17)*. 163–171.

[7] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 1724–1734.

[8] Gilles Degottex, John Kane, Thomas Drugman, Tuomo Raitio, and Stefan Scherer. 2014. COVAREP: A Collaborative Voice Analysis Repository for Speech Technologies. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. 960–964.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[10] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. 2019. Learning Discrete Structures for Graph Neural Networks. In *International conference on machine learning*. http://arxiv.org/abs/1903.11960

[11] Sichao Fu, Weifeng Liu, Weili Guan, Yicong Zhou, Dapeng Tao, and Changsheng Xu. 2021. Dynamic Graph Learning Convolutional Networks for Semi-Supervised Classification. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 1s, Article 4 (mar 2021), 13 pages. https://doi.org/10.1145/3412846

[12] Difei Gao, Ke Li, R. Wang, S. Shan, and X. Chen. 2020. Multi-Modal Graph Neural Network for Joint Reasoning on Vision and Scene Text. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 12743–12753.

[13] Dimitris Gkoumas, Qiuchi Li, C. Lioma, Yijun Yu, and Da wei Song. 2021. What makes the difference? An empirical comparison of fusion strategies for multimodal language analysis. *Information Fusion* 66 (2021), 184–197.

[14] M W Goudreau, C L Giles, S T Chakradhar, and . Chen, D. 1994. First-order versus second-order single-layer recurrent neural networks. *IEEE Transactions on Neural Networks* 5, 3 (1994), 511–513.

[15] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*. 369–376.

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[17] Devamanyu Hazarika, R. Zimmermann, and Soujanya Poria. 2020. MISA: Modality-Invariant and -Specific Representations for Multimodal Sentiment Analysis. *Proceedings of the 28th ACM International Conference on Multimedia* (2020).

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[19] Sepp Hochreiter and J1rgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[20] Ming Hou, Jiajia Tang, Jianhai Zhang, Wanzeng Kong, and Qibin Zhao. 2019. Deep Multimodal Multilinear Fusion with High-order Polynomial Pooling. In *Advances in Neural Information Processing Systems*. 12113–12122.

[21] Feiran Huang, Kaimin Wei, Jian Weng, and Zhoujun Li. 2020. Attention-Based Modality-Gated Networks for Image-Text Sentiment Analysis. *ACM Trans. Multimedia Comput. Commun. Appl.* 16, 3, Article 79 (jul 2020), 19 pages. https://doi.org/10.1145/3388861

[22] Rongrong Ji, Fuhai Chen, Liujuan Cao, and Yue Gao. 2019. Cross-Modality Microblog Sentiment Prediction via Bi-Layer Multimodal Hypergraph Learning. *IEEE Transactions on Multimedia* 21, 4 (2019), 1062–1075. https://doi.org/10.1109/TMM.2018.2867718

[23] Onno Kampman, Elham J. Barezi, Dario Bertero, and Pascale Fung. 2018. Investigating Audio, Visual, and Text Fusion Methods for End-to-End Automatic Personality Prediction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

[24] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*.

[25] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *Proceedings of International Conference on Learning Representations*.

[26] Qiuchi Li, Dimitris Gkoumas, Christina Lioma, and Massimo Melucci. 2021. Quantum-inspired multimodal fusion for video sentiment analysis. *Information Fusion* 65 (2021), 58–71.

[27] Paul Pu Liang, Zhun Liu, Yao-Hung Hubert Tsai, Qibin Zhao, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2019. Learning Representations from Imperfect Time Series Data via Tensor Rank Regularization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1569–1576.

[28] Paul Pu Liang, Ziyin Liu, Amir Zadeh, and Louis Philippe Morency. 2018. Multimodal Language Analysis with Recurrent Multistage Fusion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 150–161.

[29] Zhun Liu, Ying Shen, Paul Pu Liang, Amir Zadeh, and Louis Philippe Morency. 2018. Efficient Low-rank Multimodal Fusion with Modality-Specific Factors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2247–2256.

[30] Sijie Mai, Haifeng Hu, and Songlong Xing. 2019. Divide, Conquer and Combine: Hierarchical Feature Fusion Network with Local and Global Perspectives for Multimodal Affective Computing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 481–492.

[31] Sijie Mai, Haifeng Hu, and Songlong Xing. 2020. Modality to Modality Translation: An Adversarial Representation Learning and Graph Fusion Network for Multimodal Fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 164–172.

[32] S. Mai, H. Hu, J. Xu, and S. Xing. 2020. Multi-Fusion Residual Memory Network for Multimodal Human Sentiment Comprehension. *IEEE Transactions on Affective Computing* (2020), 1–1.
[33] S. Mai, S. Xing, and H. Hu. 2020. Locally Confined Modality Fusion Network With a Global Perspective for Multimodal Human Affective Computing. *IEEE Transactions on Multimedia* 22, 1 (2020), 122–137.
[34] Sijie Mai, Songlong Xing, and Haifeng Hu. 2021. Analyzing Multimodal Language via Acoustic-and Visual-LSTM with Channel-aware Temporal Convolution Network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2021).
[35] Sijie Mai, Ying Zeng, Shuangjia Zheng, and Haifeng Hu. 2022. Hybrid contrastive learning of tri-modal representation for multimodal sentiment analysis. *IEEE Transactions on Affective Computing* (2022).
[36] Sijie Mai, Shuangjia Zheng, Yuedong Yang, and Haifeng Hu. 2021. Communicative message passing for inductive relation reasoning. *Association for the Advancement of Artificial Intelligence (AAAI)* (2021).
[37] A. Micheli. 2009. Neural Network for Graphs: A Contextual Constructive Approach. *IEEE Transactions on Neural Networks* 20, 3 (2009), p.498–511.
[38] Aashish Kumar Misraa, Ajinkya Kale, Pranav Aggarwal, and A. Aminian. 2020. Multi-Modal Retrieval using Graph Neural Networks. *ArXiv* abs/2010.01666 (2020).
[39] Behnaz Nojavanasghari, Deepak Gopinath, Jayanth Koushik, and Louis Philippe Morency. 2016. Deep multimodal fusion for persuasiveness prediction. In *Proceedings of ACM International Conference on Multimodal Interaction*. 284–288.
[40] David Olson. 1977. From Utterance to Text: The Bias of Language in Speech and Writing. *Harvard Educational Review* 47, 3 (1977), 257–281.
[41] Ashutosh Pandey and DeLiang Wang. 2019. TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 6875–6879.
[42] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 1532–1543.
[43] Hai Pham, Paul Pu Liang, Thomas Manzini, Louis Philippe Morency, and Poczos Barnabas. 2019. Found in Translation: Learning Robust Joint Representations by Cyclic Translations Between Modalities. In *Thirty-Third AAAI Conference on Artificial Intelligence*. 6892–6899.
[44] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis Philippe Morency. 2017. Context-Dependent Sentiment Analysis in User-Generated Videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 873–883.
[45] Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Amir Hussain. 2016. Convolutional MKL Based Multimodal Emotion Recognition and Sentiment Analysis. In *Proceedings of IEEE International Conference on Data Mining (ICDM)*. 439–448.
[46] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80.
[47] Xiangbo Shu, Liyan Zhang, Yunlian Sun, and Jinhui Tang. 2020. Host–parasite: graph LSTM-in-LSTM for group activity recognition. *IEEE transactions on neural networks and learning systems* 32, 2 (2020), 663–674.
[48] Jinhui Tang, Xiangbo Shu, Zechao Li, Yu-Gang Jiang, and Qi Tian. 2019. Social anchor-unit graph regularized tensor completion for large-scale image retagging. *IEEE transactions on pattern analysis and machine intelligence* 41, 8 (2019), 2027–2034.
[49] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal Transformer for Unaligned Multimodal Language Sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 6558–6569.
[50] Yao-Hung Hubert Tsai, Martin Ma, Muqiao Yang, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Multimodal Routing: Improving Local and Global Interpretability of Multimodal Language Analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 1823–1833.
[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
[52] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of International Conference on Learning Representations*.
[53] Shiping Wang and Wenzhong Guo. 2017. Sparse Multigraph Embedding for Multimodal Feature Representation. *IEEE Transactions on Multimedia* 19, 7 (2017), 1454–1466. https://doi.org/10.1109/TMM.2017.2663324
[54] Yansen Wang, Ying Shen, Zhun Liu, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. 2019. Words Can Shift: Dynamically Adjusting Word Representations Using Nonverbal Behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7216–7223.
[55] Martin Wollmer, Felix Weninger, Tobias Knaup, Bjorn Schuller, Congkai Sun, Kenji Sagae, and Louis Philippe Morency. 2013. YouTube Movie Reviews: Sentiment Analysis in an Audio-Visual Context. *IEEE Intelligent Systems* 28, 3 (2013), 46–53.
[56] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks?. In *Proceedings of International Conference on Learning Representations*.

[57] Jianing Yang, Yongxin Wang, Ruitao Yi, Yuying Zhu, Azaan Rehman, Amir Zadeh, Soujanya Poria, and Louis-Philippe Morency. 2020. MTGAT: Multimodal Temporal Graph Attention Networks for Unaligned Human Multimodal Language Sequences. *arXiv preprint arXiv:2010.11985* (2020).

[58] Kaicheng Yang, Hua Xu, and Kai Gao. 2020. CM-BERT: Cross-Modal BERT for Text-Audio Sentiment Analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*. 521–528.

[59] Xiaocui Yang, Shi Feng, Daling Wang, and Yifei Zhang. 2020. Image-text Multimodal Emotion Classification via Multi-view Attentional Network. *IEEE Transactions on Multimedia* (2020), 1–1. https://doi.org/10.1109/TMM.2020.3035277

[60] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*. 4800–4810.

[61] Fisher Yu and Vladlen Koltun. 2016. Multi-scale context aggregation by dilated convolutions. In *Proceedings of International Conference on Learning Representations*.

[62] Hao Yuan and Shuiwang Ji. 2020. StructPool: Structured Graph Pooling via Conditional Random Fields. In *Proceedings of International Conference on Learning Representations*.

[63] Jiahong Yuan and Mark Liberman. 2008. Speaker identification on the SCOTUS corpus. *Acoustical Society of America Journal* 123 (2008), 3878. https://doi.org/10.1121/1.2935783

[64] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis Philippe Morency. 2017. Tensor Fusion Network for Multimodal Sentiment Analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1114–1125.

[65] Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis Philippe Morency. 2018. Memory Fusion Network for Multi-view Sequential Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1)*. 5634–5641.

[66] Amir Zadeh, Paul Pu Liang, Jonathan Vanbriesen, Soujanya Poria, Edmund Tong, Erik Cambria, Minghai Chen, and Louis Philippe Morency. 2018. Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2236–2246.

[67] Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis Philippe Morency. 2016. Multimodal Sentiment Intensity Analysis in Videos: Facial Gestures and Verbal Messages. *IEEE Intelligent Systems* 31, 6 (11 2016), 82–88.

[68] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*. 5165–5175.

[69] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[70] Sicheng Zhao, Shangfei Wang, Mohammad Soleymani, Dhiraj Joshi, and Qiang Ji. 2019. Affective Computing for Large-Scale Heterogeneous Multimedia Data: A Survey. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 3s, Article 93 (dec 2019), 32 pages. https://doi.org/10.1145/3363560