Hierarchical Multimodal Fusion Network with Dynamic Multi-task Learning

Tianyi Wang, Shu-Ching Chen
Knight Foundation School of Computing and Information Sciences
Florida International University
Miami, Florida
{wtian002,chens}@cs.fiu.edu

***Abstract*—Real-world data often contain multiple modalities and non-exclusive labels. Multimodal fusion is a vital step in multimodal learning that integrates features from various modalities in the vector space so that the classifier could utilize the fused vector to generate the final prediction score. Common multimodal fusion approaches rarely consider the cross-modality interactions which play an essential role in exploiting the inter-modality relationship and subsequently creating the joint modality embedding. In this paper, we propose a hierarchical multimodal fusion framework with dynamic multi-task learning. It focuses on modeling the joint embedding space for all cross-modality interactions and adjusting the task loss for optimal performance. The proposed model uses a novel hierarchical multimodal fusion network that learns the cross-modal interactions among all combinations of modalities and dynamically allocates the weights for each pair in a sample-aware fashion. Furthermore, a novel dynamic multi-task learning approach is applied to handle the multi-label problems by automatically adjusting the learning progress on both task level and sample level. We show that the proposed framework outperforms the baselines and some of the state-of-the-art methods. We also demonstrate the flexibility and modularity of the proposed hierarchical multimodal fusion and dynamic multi-task learning units, which can be applied to various types of networks.**

***Index Terms*—hierarchical multimodal fusion, graph fusion, multi-task learning**

I. Introduction

Multimodal learning has attracted great interest from the research community due to its benefit in utilizing the huge amount of real-world data which often contain multiple data sources [1] [2] [3]. Compared to its single modality counterpart, multimodal learning is the technique that focuses on exploiting the rich information underlying various input modalities. One essential step in multimodal learning is multimodal fusion, where the input features of each modality are combined to form a single vector. Therefore, the way features are fused holds substantial impact on the model's effectiveness in harvesting the information provided by multiple input sources. Contrary to traditional belief, merely increasing the number of input modalities does not always yield better results [4]. The main cause that leads to the subpar performance is due to the oversight of cross-modality interactions.

How to effectively fuse the representations of diverse modalities has become a pressing issue in multimodal learning and therefore attracted great attention of the research community. The heterogeneity nature of multimodal data creates an emerging barrier in harnessing comprehensive information across all modalities, which is the key to fully understand and utilize the rich multimedia information [5]. Early attempts on multimodal fusion tend to work on each modality separately. Each modality is trained on its own network with the resulting intermediate features combined in different stages of the processing chain, such as early fusion and late fusion [6]. However, due to the heterogeneity nature of multimodal data and the disconnection among networks, the fused vector still falls short on representing the complex distribution among modalities.

Multi-task learning (MTL) is a technique that has been quite popular in machine learning/deep learning, multi-label learning, and multi-output regression domains [7]. MTL takes advantage of the broader coverage of various domains by training multiple tasks simultaneously. MTL has been functioning remarkably well in many scenarios since a more generalized and robust model can be learned. This is achieved by sharing the knowledge among tasks, as well as a lowered chance of overfitting. An open topic in MTL is how to balance the training progress among tasks. A common practice is to assign equal weights for all tasks or to heuristically weight the training loss of each task. The former solution often yields inferior results when one task dominates the training process with an excessive loss, which can contribute to the loss function itself or the task complexity [8]. The latter solution completely relies on human judgment, which lacks the flexibility on applying to different problem domains and it usually requires tedious weight tuning process.

In this paper, we propose a novel hierarchical multimodal fusion network with dynamic multi-task learning. The multimodal fusion network hierarchically joins each modality to form a graph structure where the vertices represent joined modalities and the edges contain the cross-modality interactions. The relative importance among joined modalities in the same level is learned on a sample to sample fashion and applied to formulate the joint embedding that will be used in the next level. We also propose a dynamic multi-task learning approach that disintegrates the multi-label classification problem into various single-label binary classification tasks. By monitoring the training complexity in each task, the dynamic multi-task learning unit automatically adjusts the weighting of the task loss so that the optimal weight balance

can be achieved. The dynamic multi-task learning unit also assigns a set of initial task loss weights at the beginning of the training cycles and keeps updating them throughout the training process to ensure the task loss weights are not caught in the local minimum/maximum.

In summary, the major contributions of this paper are listed below:

- We propose a novel hierarchical multimodal fusion network that exploits the cross-modal interactions.
- A novel dynamic multi-task learning approach that automatically optimizes the model training process based on both task level and sample level training complexities. It also re-balances the loss weights for each task at the onset of the training cycles to minimize the chance of task weights being caught in the local minimum/maximum.

The remainder of this paper is organized as follows. In Section II, the literature in multimodal fusion and multi-task learning is briefly discussed. Section III provides a detailed discussion about the proposed hierarchical multimodal multi-task learning framework. Section IV presents the experimental setup and results. Finally, in Section V, we summarize the paper by discussing the key components and contributions.

II. Related Work

Traditional Multimodal fusion often operates on three levels: early fusion, late fusion, and hybrid fusion. Early fusion is usually implemented by concatenating the raw or pre-processed features of each modality immediately after the feature extraction stage [9] [10] [11]. Early fusion is simple to implement and requires a less complex network structure. However, early fusion will encounter issues when one input modality is a continuous data stream while another modality contains discrete data. Furthermore, as the number of modalities increases, it is significantly difficult to learn the cross-model interactions among heterogeneous features. Late fusion utilizes multiple networks to generate modality-specific prediction scores. Then it analyzes and manipulates the scores to arrive at the final decision [12] [13] [14] [15]. Late fusion offers several benefits over early fusion. First, the modality-specific networks enable the model to learn different semantic representations from each modality. Second, it takes advantage of the domain-specific models and algorithms, such as applying the convolutional neural network (CNN) based models on visual data, or recurrent neural network (RNN) based models on sequential data. However, late fusion omits the feature-level cross-modality interactions. This will lead to the loss of crucial inter-modality information. Hybrid fusion combines the strengths of late fusion and early fusion by transforming the raw input data into their higher-level representations to make it easier to fuse different modalities and learn the cross-modal representation [16] [17] [18].

Recently, Tensor fusion has attracted the attention of many studies. Tensor fusion tackles the heterogeneous data distribution challenge in multimodal learning by fusing each modality at the tensor level. As a result, it enables the model to learn the granularity of cross-modal interactions. Tensor fusion has demonstrated promising results in multimodal deep learning for visual question answering [19] and sentiment analysis [20]. Ben-younes *et al.* proposed a framework in order to solve the visual question answering problem [19]. They extracted features from both visual images and textual questions using GRU (Gated Recurrent Unit) and ResNet [21]. Then, features are fused using the tensor fusion approach. During the fusion process, a tensor based Tucker decomposition approach is utilized to parametrize the tensor correlation between visual and textual representations. Another work by Zhao *et al.* [22] used a multi-agent tensor layer and convolutional fusion to capture the cross-modal interactions.

Graph-based fusion networks transform modalities and the interactions among them into fusion graphs. Features from each modality are considered as vertices and the relationships between them are implemented as the edges. Zadeh *et al.* tried to use a Dynamic Fusion Graph (DFG) to model the n-modal dynamics [23]. Compared to Tensor Fusion, DFG achieves better training efficiency where much fewer learnable parameters are introduced. It also uses learned parameters to control the activation of certain edges, and thus dynamically changes the network structure. Multimodal metrics learning and graph-based fusion are combined to measure feature similarity between modalities [24]. Chen *et al.* [25] proposed a heterogeneous graph-based fusion network that focuses on the fusion of multimodal data with missing modalities. It uses a graph network to project the missing data with other modalities into a joint embedding space.

Multi-task learning provides several benefits by training multiple tasks simultaneously. Besides the obvious advantage of the shortened training time by performing only one training pass, it also helps the model learn a more generalized representation of the entire problem domain. This greatly reduces the chance of overfitting [7]. Multi-task learning also shows a great potential in the multimodal learning domain. Sener *et al.* [26] utilized multi-objective optimization to find the Pareto optimal solution to minimize the weighted combination of task losses. Vandenhende *et al.* [27] applied a multi-modal distillation unit to model task correlation from various levels of the network. A more recent work by Hu and Singh [28] employed an encoder-decoder mechanism to encode each input modality and decode them into a shared embedding space.

III. Methodology

A. Architecture Design

In this section, we present the architecture design of the hierarchical multimodal fusion multi-task learning framework. The framework is composed of two main components: a Hierarchical Graph Fusion Network (HGFN) and a dynamic MTL (DMTL) module. In the first step, the feature representations of each modality are fused by the HGFN. In step two, the joint feature produced in step one will be used by the DMTL module to dynamically adjust the training progress on each task.

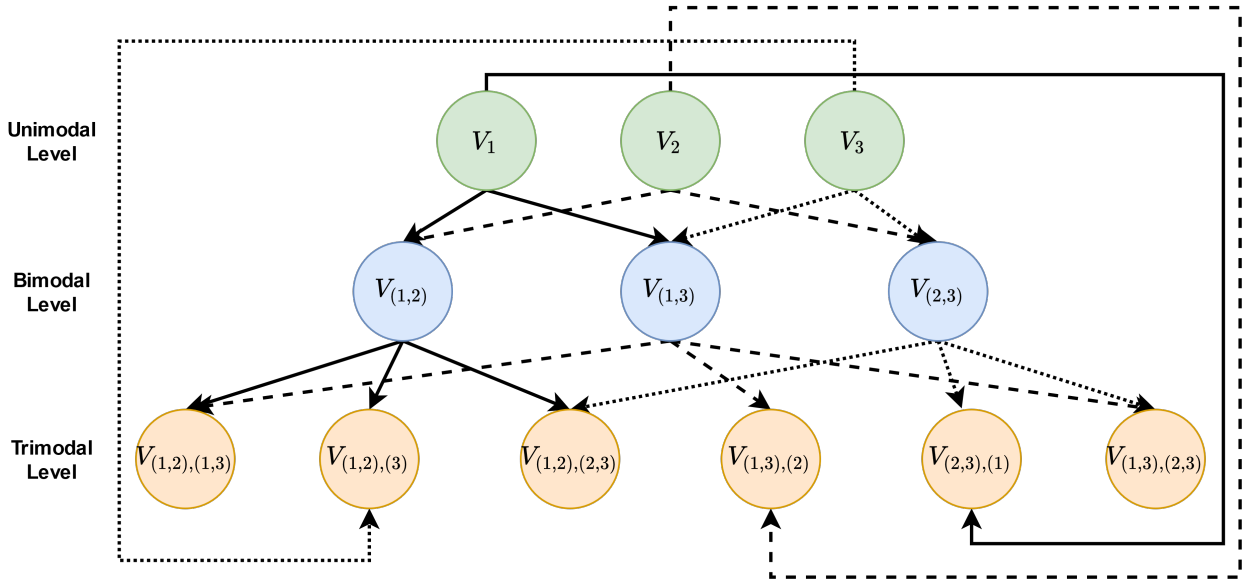

Fig. 1: Hierarchical Graph Fusion Network (HGFN) with 3 input modalities

B. Hierarchical Graph Fusion Network

Inspired by [29], we form the HGFN by exploiting the n-modal interactions. HGFN combines all modalities on unimodal, bimodal, and trimodal levels and models the interactions and relationships between each pair of combinations. An overview of the HGFN is shown in Figure 1.

The first level contains all unimodal and their interactions. We define the unimodal input feature vector V_i, where M is the total number of modalities and $i = [1, M]$. Although HGFN can be applied to any number of modalities, here we consider $M = 3$ in the rest of the paper. To model the relative importance of each modality and assign weights to the edges, we apply a Dynamic Attention Unit (DAU) to learn the importance of each modality and assign it as the weights of the connected edges. More specifically, features from each modality are first concatenated together and then pass to a network composed of 2 convolutional layers with 5 by 5 and 1 by 1 kernel size and LeakyRelu activation. Padding is performed to ensure the features of each modality have the same dimension. This process can be described as follows.

$$w_1 \oplus w_2 ... \oplus w_M = DAU(V_1 \oplus V_2 ... \oplus V_M) \tag{1}$$

where $\oplus$ is the concatenation operation, $V_1, V_2, ...V_M$ are the unimodal vectors of the M modalities, and $w_1, w_2, ...w_M$ are the corresponding weights. DAU learns the dynamic of importance score that should be assigned to each vector in a sample-based fashion. Such importance score will be used as the foundation to form the edge weights in higher levels.

In the next step, the final unimodal level vector can be obtained as the weighted average of vectors from all unimodal level vertices:

$$F_{unimodal} = \frac{1}{M} \sum_{i=1}^{M} w_i \cdot V_i \tag{2}$$

where $F_{unimodal}$ is the combined unimodal vector.

In the bimodal level, each pair of unimodal vectors are combined to form the vertices in this level. A neural network $CONV$ with one 1D convolutional layer and one dense layer with LeakyRelu activation is used to combine the unimodal vectors and produce all bimodal level vertices. This procedure can be described as:

$$\begin{gathered} V_{(a,b)} = CONV(V_a \oplus V_b) \\ a = 1, 2, ...M; b = 1, 2, ...M; a \neq b \end{gathered} \tag{3}$$

where $V_{(a,b)}$ is the bimodal vector. Regarding the edges that connect the vertices between unimodal and bimodal levels, we assume that the closer the two features in the vector space, the more homogeneous the information they possess. Therefore, the combination of such two features will not provide as much information as two distinct features do. Based on this assumption, we calculate the similarity between each pair of vertices at the bimodal level. The calculation can be described as:

$$S_{a,b} = COS(\widetilde{V}_a, \widetilde{V}_b) \tag{4}$$

where $S_{a,b}$ represents the similarity score between vertices a and b, COS is the cosine similarity function, and $\widetilde{V}_a$ and $\widetilde{V}_b$ are the softmax normalized form of vector V_a and V_b. The purpose of softmax normalization is to constrain the values of both vectors to be between 0 and 1. According to our assumption, the more similar two vectors are, the less weight they should carry when combined. In other word, the edge weight between the two vertices should grow in inverse proportion to the similarity score. Therefore, the edge weight that connects vertex a in the unimodal level and vertex ab in the bimodal level is calculated as $\frac{w_a}{S_{a,b}+\theta}$. Similarly, the edge weight that connects vertex b and ab is defined as $\frac{w_b}{S_{a,b}+\theta}$. Term θ is an adjustable factor that controls the growth rate

with a value between 0 and 1. Based on the empirical study, $\theta = 0.5$ is used in this paper. Consequently, the vertex weight in the bimodal level is formulated as:

$$q_{a,b} = \frac{w_a + w_b}{S_{a,b} + \theta}$$
$$w_{a,b} = \frac{e^{q_{a,b}}}{\sum_{j=1}^{M} \sum_{k=1, j \neq k}^{M} e^{q_{j,k}}} \tag{5}$$

where $q_{a,b}$ is the vertex weight for $V_{(a,b)}$ in the bimodal level, and $w_{a,b}$ represents the softmax normalized form of $q_{a,b}$. Then, the final combined bimodal level vector can be described as:

$$F_{bimodal} = \sum_{a=1}^{M} \sum_{b=1, a \neq b}^{M} w_{a,b} \cdot V_{(a,b)} \tag{6}$$

where $F_{bimodal}$ is the combined bimodal vector

In the trimodal level, all calculations are similar to the procedure illustrated in the bimodal part. Equations (4), (5), and (6) are used to calculate the trimodal level similarity scores, vertex weight, and combined trimodal vector. The trimodal level contains two types of vertices: 1) the combination of bimodal vertices; and 2) each bimodal vertex is combined with the unimodal vertex that is not included in the formation of this bimodal vertex. Therefore, for a dataset with 3 input modalities, there will be a total of 6 vertices in the trimodal level.

In the last step, the combined vectors from unimodal, bimodal, and trimodal levels are concatenated to form the final combined vector $F_{combined}$:

$$F_{combined} = F_{unimomdal} \oplus F_{bimodal} \oplus F_{trimodal} \tag{7}$$

C. Dynamic Multi-tasking Learning Module

MTL calculates the final training loss as a linear combination of all task losses, which is used to optimize the model parameters. Common MTL approaches either assign equal weights to all tasks or assign each task with a different weight according to the empirical study. Based on our prior studies [8] [30], we introduce the dynamic MTL (DMTL) module that is capable of learning the task loss weights on both sample level and task level. It also re-balances the initial task loss at each training cycle to avoid the loss weights from falling in the local minimum/maximum.

DMTL on sample level aims to allocate a higher priority in learning the input samples that are misclassified. By using the Cross-Entropy loss function as an example, we can describe this process as:

$$CE(p_d) = -log(p_d) \tag{8}$$

where

$$p_d = \begin{cases} p, & if\ y = 1 \\ 1 - p, & otherwise \end{cases} \tag{9}$$

where the ground truth label is $y \in \{0, 1\}$, and $0 \leq p \leq 1$ is the probability of a sample to be labeled as 1. The sample level loss weighting function L_s is defined as:

$$L_s(x) = -(1 - p_d)^{\beta} \log(p_d) \tag{10}$$

where x is the input data and β is the sample level focusing parameter that controls the magnitude of weight reduction for easy (true negative) samples. When p_d is small and the sample is misclassified, the value of $(1 - p_d)^{\beta}$ is closer to 1, which has very limited impact on the loss. On the other hand, as the value of p_d increases, $(1 - p_d)^{\beta}$ gradually becomes 0, which down-weights the loss produced by correctly classified samples. This forces the model to allocate more resources on learning hard (false negative) samples.

In comparison, DMTL on task level automatically adjusts the weights on losses generated by each task-specific network. The training loss of each task is monitored and used as the metric to adjust the weighted gradient in each layer. In practice, we use a custom loss function to minimize the differences between: 1) weight gradient among all tasks; and 2) the training rate weighted average gradient. The L_1 norm task level dynamic balancing (TDB) loss function L_{TDB} at training iteration t is defined as:

$$L_{TDB}(t) = \sum_{f} \frac{N}{n_f} \left| G_W^{(f)}(t) - \overline{G}_W(t) \times [r_f(t)]^{\alpha} \right|_1 \tag{11}$$

where N represents the total number of samples, n_f is the number of positive samples in task f, $\frac{N}{n_f}$ is the inverted task sample distribution ratio for task f, W represents the weight parameter of the last task-specific network layer, $G_W^{(f)}(t)$ is the L_2 norm gradient of the weighted task loss for task f at iteration t, $\overline{G}_W(t)$ is the average gradient of all tasks at iteration t, $r_f(t)$ represents the inverse training rate of task f, and term α controls the magnitude of the inverse training rate.

In some cases, the task loss weight update through TDB may not be sufficient if the task difficulty is overly skewed. This may slow down the task loss updating process and cause the loss weights of some tasks to fall into the local minimum/maximum. To resolve this issue, we re-balance the task loss weights after a complete training cycle. This ensures a more aggressive loss updating process which helps the model quickly reach the optimal task loss weight and avoid the loss weights from falling into the local minimum/maximum.

In practice, the average training losses for each task through the entire training cycle are calculated. Then, a weight scalar is generated by dividing the largest value among all average training losses with the average training loss of each task. Finally, the weight scalar is applied to all tasks to re-balance the task losses at the beginning of the training cycle. We keep track of the validation loss and the training process will stop if the validation loss stopped decreasing.

IV. Experiments

A. Datasets

CrisisMMD [31] is a multimedia Twitter dataset with more than 16,000 tweets and 18,000 images that are related to seven major natural disaster events. Each sample is labeled with 3 groups of concepts: data informative level, humanitarian category, and damage level. The data informative level represents the amount of information carried, the humanitarian category

covers the type of humanitarian crisis and relieving efforts occurred in the scene, and the damage level is the severity of damage on infrastructures and utilities. We report F1 score, Hamming Loss (HL), and Mean Average Precision (MAP) on this dataset. For F1 and MAP, the higher the score the better, whereas for HL the lower the score the better.

YouTube Disaster dataset [32] is a multi-label YouTube hurricane disaster video dataset that contains more than 1,500 video clips and the corresponding text descriptions. Each sample is manually labeled with 7 concepts based on the elements present in the scene. These concepts include demonstration, emergency response, flood/storm, human relief, damage, victim, and speak/briefing/interview. We report the model performance in F1 score, Hamming Loss, and Mean Average Precision on this dataset as well.

TABLE I: Data informative concept performance evaluation on the CrissMMD dataset

Method	F1	HL	MAP
CFC + LSEW	0.623	0.237	0.587
MATF + LSEW	0.774	0.151	0.738
GFN + LSEW	0.813	0.104	0.762
HGFN + LSEW	0.839	0.097	0.794
CFC + MOO	0.685	0.202	0.638
CFC + MTI-NET	0.673	0.214	0.625
CFC + DMTL	0.736	0.164	0.709
HGFN + DMTL	0.862	0.041	0.825

TABLE II: Humanitarian category concept performance evaluation on the CrissMMD dataset

Method	F1	HL	MAP
CFC + LSEW	0.527	0.293	0.496
MATF + LSEW	0.681	0.207	0.649
GFN + LSEW	0.677	0.209	0.642
HGFN + LSEW	0.712	0.181	0.695
CFC + MOO	0.603	0.246	0.571
CFC + MTI-NET	0.614	0.237	0.588
CFC + DMTL	0.686	0.194	0.660
HGFN + DMTL	0.762	0.153	0.749

TABLE III: Damage level concept performance evaluation on the CrissMMD dataset

Method	F1	HL	MAP
CFC + LSEW	0.634	0.229	0.607
MATF + LSEW	0.781	0.148	0.745
GFN + LSEW	0.819	0.117	0.793
HGFN + LSEW	0.852	0.080	0.839
CFC + MOO	0.693	0.181	0.664
CFC + MTI-NET	0.688	0.186	0.650
CFC + DMTL	0.746	0.149	0.715
HGFN + DMTL	0.913	0.029	0.897

B. Experimental Setup

Visual Feature Extraction: We use ImageNet [33] pre-trained Inception V3 [34] model as the feature extractor for the visual data. Regarding the YouTube Disaster dataset, each video clip is subsampled into 40 frames and resized and cropped into 224 by 224 pixels.

TABLE IV: Performance evaluation on the YouTube Disaster dataset

Method	F1	HL	MAP
CFC + LSEW	0.769	0.157	0.722
MATF + LSEW	0.865	0.053	0.818
GFN + LSEW	0.889	0.041	0.805
HGFN + LSEW	0.931	0.024	0.890
CFC + MOO	0.874	0.040	0.828
CFC + MTI-NET	0.882	0.035	0.831
CFC + DMTL	0.922	0.027	0.904
HGFN + DMTL	0.987	0.011	0.958

Textual Feature Extraction: Embeddings from Language Models (ELMo) representation [35] is used to generate the word embedding for textual data. Compared to traditional text embedding techniques such as Word2vec [36] and Glove [37], ELMo can capture the morphological information and also excel in handling out of vocabulary words.

Audio Feature Extraction: A pre-trained SoundNet [38] is used to extract the audio features.

For the CrissMMD dataset, features generated by each pre-trained model are directly passed to HGFN to perform multimodal fusion. To exploit the temporal information in the YouTube Disaster dataset, features generated by the pre-trained models are first fed into a small neural network with 2 Bidirectional Gated Recurrent Unit (Bi-GRU) layers with attention enabled. Then, the intermediate vectors are processed by HGFN, which is similar to the process applied to CrissMMD.

For both datasets, 60% of the data is used for training, 20% for validation, and 20% for testing. The validation set is used to tune all hyperparameters, and the term α in the TDB loss function is set to 1 based on the empirical study. Adam [39] is used for optimizing the training process and the initial learning rate is set to 0.01.

The DMTL module is applied to 3 concept groups of the CrissMMD dataset, in which each concept is modeled as a distinct task. In comparison, we consider each label in the YouTube Disaster dataset as a single task. This converts the original multi-label classification problem into an MTL problem.

C. Results and Discussion

Several baselines including the state-of-the-art methods are selected to demonstrate the performance of our proposed framework. The multimodal fusion baselines include : 1) a common fuse by concatenation (CFC) approach that simply concatenates each modality immediately after the initial feature extraction step; 2) tensor-based fusion method MAFT [22]; and 3) Graph Fusion Network (GFN) [29]. The baselines for MTL include: 1) a linear sum of all task loss with equal weights (LSEW); 2) Multi-Objective Optimization (MOO) [26]; and 3) Multi-scale Task Interaction NETwork (MTI-NET) [27]. For comparison purposes, we replace the

TABLE V: Per-concept classification accuracy on YouTube Disaster dataset

Approach	Demonstration	Emergency Response	Flood/Storm	Human Relief	Damage	Victim	Briefing
CFC + LSEW	0.823	0.812	0.866	0.829	0.787	0.780	0.875
MATF + LSEW	0.853	0.841	0.897	0.854	0.811	0.804	0.905
GFN + LSEW	0.866	0.851	0.902	0.865	0.831	0.824	0.880
HGFN + LSEW	0.914	0.909	0.960	0.927	0.913	0.895	0.923
CFC + MOO	0.841	0.835	0.887	0.853	0.819	0.804	0.890
CFC + MTI-NET	0.866	0.852	0.875	0.841	0.833	0.812	0.906
CFC + DMTL	0.933	0.908	0.932	0.931	0.942	0.903	0.915
HGFN + DMTL	0.955	0.971	0.989	0.973	0.952	0.917	0.982

model low-level layers of each baseline with the aforementioned pre-trained models.

Multimodal fusion strategies: Table I, Table II, and Table III demonstrate the performance of the proposed HGFN and DMTL approaches, as well as other baseline methods on the data informative, humanitarian category, and damage level concepts of the CrissMMD dataset. Table IV shows the experimental results on the YouTube Disaster dataset. It can be observed that for both datasets, the CFC+LSEW combination yields the lowest score in all metrics. This is not surprising since a simple concatenation of features in the early stage often fails to reflect the heterogeneous distribution of different modalities. Moreover, an equal weight linear sum of task loss in MTL has very limited effectiveness or even negative impact when a few tasks dominate the training process.

Tensor-based fusion method MATF and graph-based fusion method GFN both demonstrate performance improvements comparing to the CFC+LSEW vanilla approach. GFN exhibits a clear edge over MATF, especially on data with more input modalities, such as the YouTube Disaster dataset. This is partly due to the fact that common tensor fusion approaches like MATF only model the joint embedding representation after the fusion operation, whereas GFN fills this gap by learning the inter-modality interaction during the early stage.

Our proposed HGFN outperforms all baselines and beat the 2nd best performer by 4.2% in F1 score and 8.5% in MAP. We argue that this can partly be contributed to the DAU that learns the relative importance of each modality and integrates it at the very beginning of the graph fusion network. We also report the per-concept results on the YouTube Disaster dataset in Table V which shows the classification accuracy of all 7 concepts. It can be observed that our proposed approach outperforms GFN+LSEW (the second best result) by up to 8.2% in the damage concept. Our model exhibits consistent performance on both datasets regarding the multimodal fusion results.

Multi-task learning strategies: Table I, Table II, Table III, and Table IV also illustrate the results of MTL methods on both CrissMMD and YouTube Disaster datasets. Both MOO and MTI-NET exhibit stronger performance comparing to the equal weight linear sum MTL approach. However, the overall improvement is not quite significant. A probable explanation is in the situation of severe class imbalance, where there will be a substantial performance hit on both methods. Our proposed approach handles the class imbalance issue by introducing the inverted task sample distribution ratio term in the DMTL loss function, which helps the model further penalize the majority classes by allocating more resources to the minority classes. We also argue that re-balancing the task loss weight at the beginning of a training cycle helps our model continue reducing the total training loss; while this mechanism is absent in the other two methods.

For the CrissMMD dataset, our proposed DMTL approach outperforms the 2nd best method by 7.2% in F1 score and 7.3% in MAP. Regarding the YouTube Disaster dataset, our approach also leads the 2nd best performer in classification accuracy by 9.1% in the victim concept.

Overall, Our proposed model with hierarchical graph fusion network and dynamic MTL achieves the best performance among all baselines in both CrissMMD and YouTube Disaster datasets. Furthermore, the modularity design of HGFN and DMTL module makes it very flexible and easy to apply to other data types and model structures.

V. Conclusion

In this paper, we propose a hierarchical multimodal multi-task learning framework that learns the joint embedding space for all cross-modality interactions and handles input data with multiple non-exclusive labels. We first analyzed the challenges of multimodal fusion and designed a novel hierarchical graph fusion network that is capable of exploiting joint embedding among all cross-modality interactions. The proposed HGFN first produces the importance score for each unimodal vertice and utilizes it to derive the interactions among bimodal and trimodal vertices. Furthermore, we introduced a novel DMTL module that automatically adjusts the loss weight for each task based on their learning complexity. The DMTL module also takes into account the sample difficulty factors by allocating more resources to the hard samples. A task loss weight re-balancing mechanism is in place to ensure an optimal weight distribution at the beginning of the training cycle, which effectively prevents the weight from falling into the local minimum/maximum. Experimental results on two multimedia datasets show that our method outperforms baseline approaches by a clear margin. Moreover, our proposed framework can be applied to other data domains and network structures with little effort due to its modular nature.

Acknowledgment

This research is partially supported by NSF CNS-1952089.

REFERENCES

[1] S.-C. Chen and R. L. Kashyap, "A spatio-temporal semantic model for multimedia database systems and multimedia information systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 4, pp. 607–622, 2001.

[2] S.-C. Chen, M.-L. Shyu, M. Chen, and C. Zhang, "A decision tree-based multimodal data mining framework for soccer goal detection," in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, 2004, pp. 265–268.

[3] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. Iyengar, "Multimedia big data analytics: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–34, 2018.

[4] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[5] P. Hu, Y.-A. Huang, K. C. Chan, and Z.-H. You, "Learning multi-modal networks from heterogeneous data for prediction of lncrna–mirna interactions," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 5, pp. 1516–1524, 2019.

[6] C. G. Snoek, M. Worring, and A. W. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, 2005, pp. 399–402.

[7] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[8] T. Wang and S.-C. Chen, "Multi-label multi-task learning with dynamic task weight balancing," in *IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, 2020, pp. 245–252.

[9] X. Hu, K. Yang, L. Fei, and K. Wang, "Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation," in *IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 1440–1444.

[10] Y. Sun, W. Zuo, and M. Liu, "Rtfnet: Rgb-thermal fusion network for semantic segmentation of urban scenes," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2576–2583, 2019.

[11] L. Deng, M. Yang, T. Li, Y. He, and C. Wang, "Rfbnet: deep multimodal networks with residual fusion blocks for rgb-d semantic segmentation," *arXiv preprint arXiv:1907.00135*, 2019.

[12] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4644–4651.

[13] Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang, "Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3029–3037.

[14] Y. Zhang, O. Morel, M. Blanchon, R. Seulin, M. Rastgoo, and D. Sidibé, "Exploration of deep learning-based multimodal fusion for semantic road scene segmentation." in *VISIGRAPP (5: VISAPP)*, 2019, pp. 336–343.

[15] T. Meng, L. Lin, M.-L. Shyu, and S.-C. Chen, "Histology image classification using supervised classification and multimodal fusion," in *IEEE International Symposium on Multimedia*, 2010, pp. 145–152.

[16] D. Lin, G. Chen, D. Cohen-Or, P.-A. Heng, and H. Huang, "Cascaded feature network for semantic segmentation of rgb-d images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1311–1319.

[17] J. Jiang, L. Zheng, F. Luo, and Z. Zhang, "Rednet: Residual encoder-decoder network for indoor RGB-D semantic segmentation," *CoRR*, vol. abs/1806.01054, 2018.

[18] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *International Journal of Computer Vision*, pp. 1–47, 2019.

[19] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, "Mutan: Multimodal tucker fusion for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2612–2620.

[20] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L. Morency, "Tensor fusion network for multimodal sentiment analysis," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2017, pp. 1103–1114.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2016, pp. 770–778.

[22] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 126–12 134.

[23] A. B. Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency, "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 (Volume 1: Long Papers)*, 2018, pp. 2236–2246.

[24] M. Angelou, V. Solachidis, N. Vretos, and P. Daras, "Graph-based multimodal fusion with metric learning for multimodal classification," *Pattern Recognition*, vol. 95, pp. 296–307, 2019.

[25] J. Chen and A. Zhang, "Hgmf: Heterogeneous graph-based fusion for multimodal data with incompleteness," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1295–1305.

[26] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Annual Conference on Neural Information Processing Systems*, 2018, pp. 525–536.

[27] S. Vandenhende, S. Georgoulis, and L. Van Gool, "Mti-net: Multi-scale task interaction networks for multi-task learning," in *European Conference on Computer Vision*. Springer, 2020, pp. 527–543.

[28] R. Hu and A. Singh, "Unit: Multimodal multitask learning with a unified transformer," *arXiv preprint arXiv:2102.10772*, 2021.

[29] S. Mai, H. Hu, and S. Xing, "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, no. 01, 2020, pp. 164–172.

[30] T. Wang and S.-C. Chen, "Multi-label multi-task learning with dynamic task weight balancing," in *IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, 2020, pp. 245–252.

[31] F. Alam, F. Ofli, and M. Imran, "Crisismmd: Multimodal twitter datasets from natural disasters," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, no. 1, 2018.

[32] S. Pouyanfar, T. Wang, and S.-C. Chen, "A multi-label multimodal deep learning framework for imbalanced data classification," in *IEEE conference on multimedia information processing and retrieval (MIPR)*, 2019, pp. 199–204.

[33] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2009, pp. 248–255.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[35] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT)*. Association for Computational Linguistics, 2018, pp. 2227–2237.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., 2013.

[37] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2014, pp. 1532–1543.

[38] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, 2016, pp. 892–900.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR)*, 2015.