Classifier-guided Gradient Modulation for Enhanced Multimodal Learning

Zirun Guo[1,2], **Tao Jin**[1†], **Jingyuan Chen**[1], **Zhou Zhao**[1,2]

[1] Zhejiang University, [2] Shanghai AI Lab
zrguo.cs@gmail.com

Abstract

Multimodal learning has developed very fast in recent years. However, during the multimodal training process, the model tends to rely on only one modality based on which it could learn faster, thus leading to inadequate use of other modalities. Existing methods to balance the training process always have some limitations on the loss functions, optimizers and the number of modalities and only consider modulating the magnitude of the gradients while ignoring the directions of the gradients. To solve these problems, in this paper, we present a novel method to balance multimodal learning with **C**lassifier-**G**uided **G**radient **M**odulation (CGGM), considering both the magnitude and directions of the gradients. We conduct extensive experiments on four multimodal datasets: UPMC-Food 101, CMU-MOSI, IEMOCAP and BraTS 2021, covering classification, regression and segmentation tasks. The results show that CGGM outperforms all the baselines and other state-of-the-art methods consistently, demonstrating its effectiveness and versatility. Our code is available at https://github.com/zrguo/CGGM.

1 Introduction

Humans perceive the world in a multimodal way, such as sight, touch and sound. These multimodal features can provide comprehensive information to help us understand and explore the environment. Recent years have witnessed great success in multimodal learning, such as visual question answering [2], multimodal sentiment analysis [18] and multimodal retrieval [26, 13].

Although multimodal learning has made significant progress in recent years, inadequate use of different modality information during training remains a challenge. Theoretically, for example, Wu et al. [25] put forward the greedy learner hypothesis which states that a multimodal model learns to rely on one of the input modalities, based on which it could learn faster, and does not continue to learn to use other modalities. Huang et al. [12] find that during joint training, multiple modalities will compete with each other and some modalities will fail in the competition. Experimentally, on some multimodal datasets, there is little improvement in accuracy between training with only one modality and training with all modalities [18, 21]. These theoretical analyses and experimental results demonstrate the inefficiency of multimodal learning to fully utilize and integrate information from different modalities.

To deal with this problem, recent studies [25, 17, 15, 8, 28, 9] investigate the training process of multimodal learning and propose gradient modulation strategies to better integrate the information of different modalities and balance the training process in some situations. However, all of these methods can not be applied easily for some limitations. For example, Wu et al. [25], Peng et al. [17], Li et al. [15] and Hua et al. [11] propose balancing methods based on cross-entropy loss for

†Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-4238

classification tasks. For regression tasks or other tasks, we can not use these strategies. Besides, most of these methods can just deal with situations where there are only two modalities. For example, Wu et al. [25] propose the conditional learning speed which is difficult to calculate and employ if there are more than two modalities. For situations where there are more modalities, these methods can not be applied. Furthermore, most of these methods only consider modulating the magnitude of the gradients while ignoring the directions of the gradients.

Based on the above observations, in this paper, we propose a novel method to balance multimodal learning with **C**lassifier-**G**uided **G**radient **M**odulation (CGGM). In CGGM, we consider a more general situation with no limitations on the type of tasks, optimizers, the number of modalities, etc. Additionally, we consider both the magnitude and directions of the gradients to fully boost the training process of multimodal learning. Specifically, we add classifiers to evaluate the utilization rate of each modality and obtain the unimodal gradients. Then, we leverage the utilization rate to adaptively modulate the magnitude of the gradients of encoders and use the unimodal gradients to instruct the model to optimize towards a better direction.

We conduct extensive experiments on four multimodal datasets: UPMC-Food 101 [23], CMU-MOSI [27], IEMOCAP [3], and BraTS 2021 [1]. UPMC-Food 101 and IEMOCAP are classification tasks, CMU-MOSI is a regression task, and BraTS 2021 is a segmentation task. CGGM outperforms all the baselines and other state-of-the-art methods, demonstrating its effectiveness and universality. In summary, our contributions are as follows:

- We propose CGGM to balance multimodal learning by both considering the magnitude and direction of the gradients.
- CGGM can be easily applied to many multimodal tasks and networks with no limitations on the type of tasks, optimizers, the number of modalities, etc. which indicates its versatility.
- Our proposed CGGM brings consistent improvements to various tasks, including classification, regression and segmentation tasks. Extensive experiments show that CGGM outperforms other state-of-the-art methods, demonstrating its effectiveness.

2 Related Work

Multimodal Learning. One of the main challenges of multimodal learning is how to effectively utilize and integrate the information from different modalities to complement each other. According to the fusion strategies, there are three main multimodal fusion strategies: early fusion, intermediate fusion and late fusion. In early fusion methods [16, 24], raw data from different modalities is combined via concatenation or other methods at the input level before being fed into a model. Intermediate fusion [14] methods combine data from different modalities at various intermediate processing stages within a model architecture. Late fusion [2, 18] methods process data from each modality independently through separate models and combine them at a later stage. In general, late fusion is the predominant method used in multimodal learning. The main reason [14] is that the architecture of each unimodal stream has been carefully designed over the years to achieve state-of-the-art performance for each modality. Therefore, we can leverage these pre-trained models [5, 6] to achieve better results. Therefore, in this paper, our method is based on late fusion.

These fusion strategies are able to integrate information from different modalities effectively, but they have limited improvements to utilize information from different modalities to complement each other. In other words, they are not able to deal with the modality competition [12] or imbalanced multimodal learning. When the dominant modality is missing [10] or corrupted, the performance would degrade significantly. Different from these fusion strategies, our method aims to make relatively full use of the information of each modality and address the imbalanced multimodal learning.

Balanced Multimodal Learning. The inefficiency in fully utilizing and integrating information from multiple modalities poses a great challenge to the multimodal learning field. Some studies [18, 21] present that there is little improvement in accuracy between training with only one modality and training with all modalities. Wang et al. [22] show that multimodal models using multiple modalities can be even inferior to those using only one modality. To balance the multimodal learning process and fully utilize different modalities, a series of balanced multimodal learning methods [25, 17, 15, 8, 28, 9, 7, 11] are proposed. Wu et al. [25] propose the conditional learning speed to capture the relative learning speed between modalities and balance the learning process. Peng et al. [17]

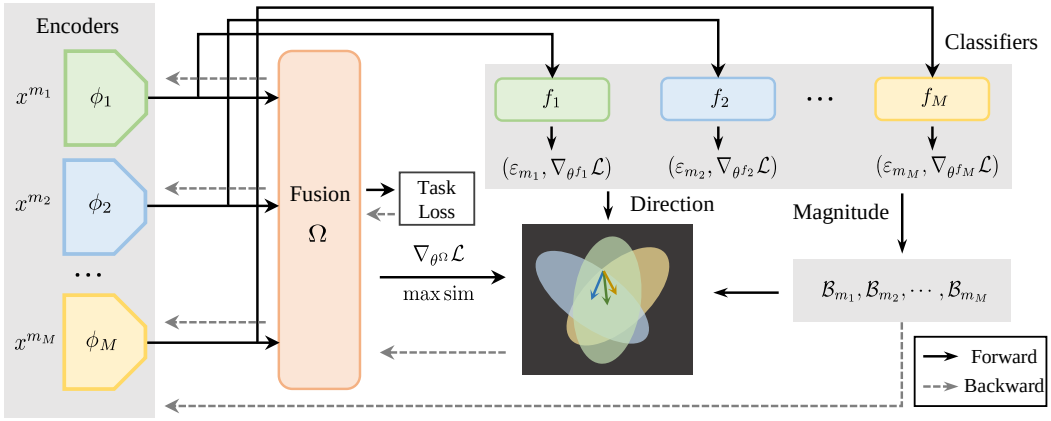


Figure 1: The overall architecture of CGGM. During the training stage, classifiers are introduced to calculate the directions of unimodal gradients and evaluation metrics. During the inference stage, the classifiers are discarded.

propose a gradient modulation strategy that adaptively controls the optimization of each modality via monitoring the discrepancy of their contribution towards the learning objective. More recently, Fan et al. [8] propose the prototypical modal rebalance strategy to introduce different learning strategies for different modalities. Li et al. [15] propose an adaptive gradient modulation method that can boost the performance of multimodal models with various fusion strategies. Hua et al. [11] dynamically adjust the learning objective with a reconcilement regularization against competition with the historical models.

However, all of these previous works have certain limitations and can only be used in some specific situations. For example, Wu et al. [25] propose conditional learning speed based on intermediate fusion strategy which makes it hard to apply to situations where there are more than two modalities or where the network is not based on intermediate fusion. Peng et al. [17], Fan et al. [8], Fu et al. [9], Li et al. [15] and Hua et al. [11] propose the balancing strategies with the assumption of the cross-entropy loss function mainly for classification. Particularly, Peng et al. [17] employ the SGD optimizer. Different from these methods, we consider a more general situation with no limitations on the number of modalities, the optimizer, the loss function and so on. Additionally, most of existing methods only consider the magnitude of the gradients and ignore the directions of the gradients. In contrast, we consider both of them.

3 Proposed Method

3.1 Problem Settings

Suppose there are M modalities, referred to as $m_1, m_2, \cdots, m_M$. We denote the multimodal dataset as $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N}$, where N is the number of data in the dataset and $\boldsymbol{x_i} = (x_i^{m_1}, x_i^{m_2}, \cdots, x_i^{m_M})$.

We consider the most common structure (Figure 1) in multimodal models, where the inputs of different modalities are first fed into modality-specific encoders and then the representations of all modalities are inputted into a fusion module. We denote the encoder of modality m_i as ϕ_i where $i = 1, 2, \cdots, M$ and the fusion module as Ω.

For the forward propagation, the features are first inputted into the encoder:

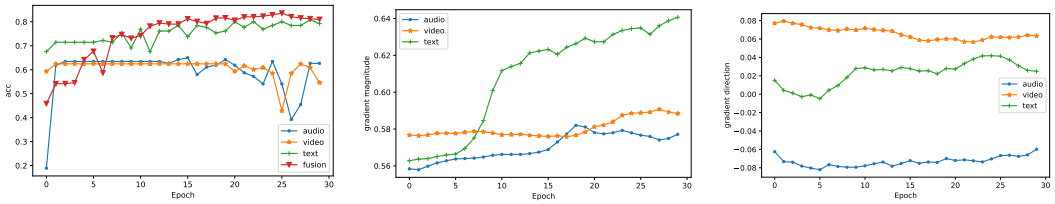
$$h_i = \phi_i(x^{m_i}) \tag{1}$$

where h_i is the representation of modality m_i. After obtaining the representations of all modalities, the fusion module is applied:

$$\hat{y} = \mathcal{F}(\Omega([h_1, h_2, \cdots, h_M])) \tag{2}$$

where $\hat{y}$ is the prediction, $[\cdots]$ is the concatenation operation, and $\mathcal{F}(\cdot)$ is the prediction head to predict the answer. $\Omega(\cdot)$ fuses the multimodal representations and outputs the fused feature as the prediction token.

https://doi.org/10.52202/079017-4238



(a) (b) (c)

Figure 2: (a) Accuracy of each modality and the fusion. (b) Gradient magnitude of each modality. We use the Euclidean norm of the gradient vector to represent the gradient magnitude. (c) Gradient direction between each modality and their fusion. We use cosine similarity to represent the direction between two gradient vectors. We get all the results on the CMU-MOSI dataset.

3.2 Gradient Analysis

To introduce CGGM, we first analyze the gradient updating process. We denote the loss function as $\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(\hat{y}_\theta^i, y^i)$ where θ represents the parameters of the network, $\hat{y}_\theta^i$ is the prediction and y^i is the ground truth. For simplicity, we use $\hat{y}^i$ to represent the predictions in the following context. Different from previous methods which only consider cross-entropy loss [17, 8, 11], our $\mathcal{L}$ can be cross-entropy loss, L1 loss or any other loss functions. With the *Gradient Descent* (GD) optimization method, the parameters of the fusion module Ω and encoders ϕ_i can be updated as:

$$\theta_{t+1}^{\Omega} = \theta_t^{\Omega} - \alpha \nabla_{\theta^{\Omega}} \mathcal{L}(\theta_t^{\Omega}) = \theta_t^{\Omega} - \alpha \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\partial \mathcal{F}}{\partial \Omega} \right)^{\top} \frac{\partial \ell\left(\hat{y}^n, y^n\right)}{\partial \mathcal{F}} \tag{3}$$

$$\theta_{t+1}^{\phi_i} = \theta_t^{\phi_i} - \alpha \nabla_{\theta^{\phi_i}} \mathcal{L}(\theta_t^{\phi_i}) = \theta_t^{\phi_i} - \alpha \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\partial \mathcal{F}}{\partial \Omega} \frac{\partial \Omega}{\partial \phi_i} \right)^{\top} \frac{\partial \ell\left(\hat{y}^n, y^n\right)}{\partial \mathcal{F}} \tag{4}$$

where α is the learning rate, N is batch size, and t is the iteration. According to the chain rule used to find the gradient in backpropagation, the update of ϕ_i will influence the update of Ω, and vice versa. According to Figure 2(a) and 2(b), the gradient and the accuracy of the dominant modality will increase during the training process while the other two remain stable. Particularly, the gradient magnitude of the text modality increases very fast during the training process. This suggests the encoder of the dominant modality will be updated much faster than others, which makes $\frac{\partial \Omega}{\partial \phi}$ much larger. This phenomenon can also be validated by previous works [8, 17, 25]. Besides, in Figure 2(c), we present the gradient direction between each modality and the fusion. We can observe that the similarity between audio modality and the multimodal fusion is less than 0, indicating that they optimize towards the opposite direction, thus hindering the gradient update for multimodal branch. Meanwhile, the similarity between text modality and the multimodal fusion is increasing, suggesting the optimization direction towards the dominant modality. With the progress of optimization, the encoder of the dominant modality can make relatively accurate predictions, which makes the fusion module Ω only depend on this modality (both magnitude and direction as mentioned above), leaving other encoders under-optimized.

3.3 Classifier-guided Gradient Modulation

3.3.1 Gradient Magnitude Modulation

As we discuss in Section 3.2, the gradient magnitude of the dominant modality increases fast during the training while the other modalities remain stable, thus being under-optimized. To balance the training process and make the fusion module benefit from all the encoders simultaneously, we propose the classifier-guided gradient modulation. Specifically, we use a modality-specific classifier to make predictions of h_i in Equation 1. We can write the process as:

$$\hat{y}_{m_i} = f_i(h_i) \tag{5}$$

where f_i is the classifier of modality m_i and $\hat{y}_{m_i}$ is the prediction only using modality m_i. The classifier f_i consists of 1-2 multi-head self-attention (MSA) layers [20] and a fully connected layer

for classification and regression tasks. And for segmentation tasks, f_i is a light decoder. After h_i is inputted into the fusion module Ω, it becomes a more high-level representation. Therefore, we use several MSA layers to make h_i more consistent with the output of the fusion module.

For a specific task, we have some evaluation metrics such as accuracy and mean absolute error. Here, we choose one of the evaluation metrics (*e.g.* accuracy for classification tasks and mean absolute error for regression tasks) and denote it as ε. For every iteration of training, we can get predictions from the classifiers. We denote the predictions as $\hat{\boldsymbol{y}}^{\boldsymbol{i}} = (\hat{y}^i_{m_1}, \hat{y}^i_{m_2}, \cdots, \hat{y}^i_{m_M})$ where i is the current iteration. Furthermore, we evaluate the task using $\hat{\boldsymbol{y}}^{\boldsymbol{i}}$ to get the evaluation metric $\boldsymbol{\varepsilon}^{\boldsymbol{i}} = (\varepsilon^i_{m_1}, \varepsilon^i_{m_2}, \cdots, \varepsilon^i_{m_M})$. Here, we use the difference between the two consecutive $\boldsymbol{\varepsilon}$ to denote the modality-specific improvement for each iteration:

$$\begin{aligned} \Delta\boldsymbol{\varepsilon}^{\boldsymbol{t+1}} &= \boldsymbol{\varepsilon}^{\boldsymbol{t+1}} - \boldsymbol{\varepsilon}^{\boldsymbol{t}} = (\Delta\varepsilon^{t+1}_{m_1}, \Delta\varepsilon^{t+1}_{m_2}, \cdots, \Delta\varepsilon^{t+1}_{m_M}) \\ &= (\varepsilon^{t+1}_{m_1} - \varepsilon^{t}_{m_1}, \varepsilon^{t+1}_{m_2} - \varepsilon^{t}_{m_2}, \cdots, \varepsilon^{t+1}_{m_M} - \varepsilon^{t}_{m_M}) \end{aligned} \tag{6}$$

where $t = 0, 1, 2, \cdots, T$ and T is the total iterations of training. Particularly, $\boldsymbol{\varepsilon}^{\mathbf{0}}$ is initialized to $\mathbf{0}$. In some multimodal datasets, only using one of the modalities can achieve good results so we can not directly use $\boldsymbol{\varepsilon}$ to measure the utilization rate of different modalities. Therefore, it is reasonable to use the difference between $\boldsymbol{\varepsilon}$ to denote the relative improvements for each iteration. Then, we define the gradient magnitude balancing term of modality m_i for the t-th iteration as follows:

$$\mathcal{B}^t_{m_i} = \rho \frac{\sum_{k=1,k\neq i}^{M} \Delta\varepsilon^t_{m_k}}{\sum_{k=1}^{M} \Delta\varepsilon^t_{m_k}} \tag{7}$$

where ρ is a scaling hyperparameter and M is the number of modalities. According to Equation 7, it is easy to find that when the performance of the model only using modality m_i improves very fast (*i.e.* $\Delta\varepsilon^t_{m_i}$ is large), $\mathcal{B}^t_{m_i}$ will be small. Similarly, when the modality m_i brings relatively limited improvements to the model (*i.e.* $\Delta\varepsilon^t_{m_i}$ is small), $\mathcal{B}^t_{m_i}$ will be large. Therefore, $\mathcal{B}^t_{m_i}$ is able to measure the relative utilization rate of these modalities and we can use $\mathcal{B}^t_{m_i}$ to modulate the magnitude of the gradient of the encoder ϕ_i. So Equation 4 can be rewritten as:

$$\begin{aligned} \theta^{\phi_i}_{t+1} &= \theta^{\phi_i}_t - \alpha\mathcal{B}^{t+1}_{m_i}\nabla_{\theta^{\phi_i}}\mathcal{L}(\theta^{\phi_i}_t) \\ &= \theta^{\phi_i}_t - \alpha\rho\frac{\sum_{k=1,k\neq i}^{M} \Delta\varepsilon^{t+1}_{m_k}}{\sum_{k=1}^{M} \Delta\varepsilon^{t+1}_{m_k}}\nabla_{\theta^{\phi_i}}\mathcal{L}(\theta^{\phi_i}_t) \end{aligned} \tag{8}$$

According to Equation 2, we know that the final predictions are closely related to h_i. Therefore, after we modulate the gradient of the corresponding encoder ϕ_i, it has an impact on the input of Ω, which in turn helps the optimization of the fusion module Ω.

3.3.2 Gradient Direction Modulation

As Wu et al. [25] discover, when the model only depends on one modality to perform well, it does not continue to learn to use other modalities. As discussed in Section 3.2, it means that this modality dominates the updating of the model. Previous works [25, 17, 15] address this problem mainly by focusing on gradient magnitude modulation. However, in Section 3.2, we find that the model is optimized towards the dominant modality. Therefore, in this subsection, we introduce a method that could modulate the direction of the gradients to balance the training process.

In general, we want to balance the optimization direction of the model when the model only relies on one modality to make predictions. Therefore, we propose to enforce the gradient direction of the model as close as possible to the weighted average gradient direction of models only using one modality. We use $\mathcal{B}^t_{m_i}$ in Equation 7 as the weight term. This ensures that when the model tends to optimize towards the dominant modality, $\mathcal{B}^t_{m_i}$ can help the model use information from other modalities. Besides, since $\mathcal{B}^t_{m_i}$ changes during the training process, this term can make a dynamic adjustment to balance the optimization directions. Concretely, we can feed one modality into the model and drop other modalities by replacing them with $\mathbf{0}$ or other fixed values during training to calculate the gradient of this modality. By this method, we can calculate the unimodal gradients for all modalities. Then, we just enforce the gradient direction of the model as close as possible to the weighted average of these unimodal gradient directions. However, this method is very complex

 https://doi.org/10.52202/079017-4238

Algorithm 1 Classifier-guided gradient modulation

Input: Training dataset $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N}$, iteration number T, the number of modalities M, model $F = (\phi_i, \Omega, \mathcal{F})$, classifiers f_i, and hyperparameters.
Initiate: $\boldsymbol{\varepsilon^p} \leftarrow \mathbf{0}$, $\boldsymbol{\varepsilon^n} \leftarrow$ Empty List, $\mathcal{L}_{gm} \leftarrow 0$, classifier gradient list L_g.
for $t = 1$ **to** T **do**
  $\mathcal{D}_t \xleftarrow{\text{Sample}} \mathcal{D}$;
  Forward propagation to get representations $\boldsymbol{h} = (h_1, h_2, \cdots, h_M)$ in Equation 1;
  for $i = 1$ **to** M **do**
    Make predictions with h_i using Equation 5;
    Calculate $\varepsilon_{m_i}^t$ and append it to $\boldsymbol{\varepsilon^n}$;
    Append $\nabla_{\theta^{f_i}} \mathcal{L}(\theta^{f_i})$ to L_g;
  end for
  $\Delta\boldsymbol{\varepsilon^t} = \boldsymbol{\varepsilon^n} - \boldsymbol{\varepsilon^p}$;
  Calculate $\mathcal{B}_{m_i}^t, i = 1, 2, \cdots, M$ using Equation 7;
  Calculate the loss $\mathcal{L} = \mathcal{L}_{task} + \lambda \mathcal{L}_{gm}$ and backward;
  Calculate $\mathcal{L}_{gm}^t$ using Equation 12;
  $\boldsymbol{\varepsilon^p} \leftarrow \boldsymbol{\varepsilon^n}$, $\mathcal{L}_{gm} \leftarrow \mathcal{L}_{gm}^t$, $\boldsymbol{\varepsilon^n} \leftarrow$ Empty List;
  Update parameters using Equation 3 and 8.
end for

during training, because in every iteration we need to drop modalities to calculate the unimodal gradients, which is time-consuming with the increase in the number of modalities.

Therefore, we propose to use the gradients of the classifiers f_i to represent the unimodal gradients. We will later demonstrate they are similar (Section 4.4 and Figure 4). Here, we take the gradient of regression tasks as an example where the output dimension is 1 so the gradient is an n-d vector. For classification tasks or other tasks where the gradient is a matrix, see Appendix A for details. Concretely, we can calculate the gradient of the classifier f_i as:

$$\nabla_{\theta^{f_i}} \mathcal{L}(\theta^{f_i}) = \frac{\partial \mathcal{L}(\theta^{f_i})}{\partial f_i} = \left[\frac{\partial \mathcal{L}(\theta^{f_i})}{\partial \theta_1^{f_i}}, \frac{\partial \mathcal{L}(\theta^{f_i})}{\partial \theta_2^{f_i}}, \cdots, \frac{\partial \mathcal{L}(\theta^{f_i})}{\partial \theta_n^{f_i}} \right]^\top \tag{9}$$

where θ^{f_i} represents the parameters of f_i. Different from $\theta_t^{f_i}$ in Equation 3 and 4 where t is the iteration, $\theta_n^{f_i}$ here represents one of the variables of θ^{f_i}. Similarly, we can calculate the gradient of the classifier $\mathcal{F}$ of the fusion module as:

$$\nabla_{\theta^{\mathcal{F}}} \mathcal{L}(\theta^{\mathcal{F}}) = \frac{\partial \mathcal{L}(\theta^{\mathcal{F}})}{\partial \mathcal{F}} = \left[\frac{\partial \mathcal{L}(\theta^{\mathcal{F}})}{\partial \theta_1^{\mathcal{F}}}, \frac{\partial \mathcal{L}(\theta^{\mathcal{F}})}{\partial \theta_2^{\mathcal{F}}}, \cdots, \frac{\partial \mathcal{L}(\theta^{\mathcal{F}})}{\partial \theta_n^{\mathcal{F}}} \right]^\top \tag{10}$$

We regard $\nabla_{\theta^{f_i}} \mathcal{L}, i = 1, 2, \cdots, M$ as the unimodal gradients and $\nabla_{\theta^{\mathcal{F}}} \mathcal{L}$ as the fusion gradients. As mentioned before, we want to make $\nabla_{\theta^{\mathcal{F}}} \mathcal{L}$ as close as possible to the weighted average direction of $\nabla_{\theta^{f_i}} \mathcal{L}, i = 1, 2, \cdots, M$. Let $\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / ||\boldsymbol{u}|| ||\boldsymbol{v}||$ denote the dot product between ℓ_2 normalized $\boldsymbol{u}$ and $\boldsymbol{v}$ (*i.e.* cosine similarity). We can enforce the gradient direction of the fusion module as close as possible to the weighted average of these unimodal gradient directions by maximizing their cosine similarity:

$$\max \sum_{i=1}^{M} \mathcal{B}_{m_i}^t \text{sim} \left(\nabla_{\theta^{\mathcal{F}}} \mathcal{L}, \nabla_{\theta^{f_i}} \mathcal{L} \right) \tag{11}$$

where t is the current iteration. We rewrite Equation 11 as a loss term:

$$\mathcal{L}_{gm}^t = \frac{1}{M} \sum_{i=1}^{M} |\mathcal{B}_{m_i}^t| - \mathcal{B}_{m_i}^t \text{sim} \left(\nabla_{\theta_t^{\mathcal{F}}} \mathcal{L}, \nabla_{\theta_t^{f_i}} \mathcal{L} \right) \tag{12}$$

The cosine similarity is a number between -1 and 1. By adding $|\mathcal{B}_{m_i}^t|$ to the loss term, we can ensure that the loss $\mathcal{L}_{gm}$ is always positive. As aforementioned, when modality m_i has limited improvement, $\mathcal{B}_{m_i}^t$ is large. Therefore, the corresponding term in $\mathcal{L}_{gm}^t$ will be large, making the optimization direction towards modality m_i, which will balance the learning process.

Then our overall loss function can be written as:

$$\mathcal{L}^t = \mathcal{L}_{task} + \lambda \mathcal{L}^t_{gm} \tag{13}$$

where $\mathcal{L}_{task}$ is the task loss function (*e.g.* cross-entropy loss, L1 loss) and λ is a trade-off between the two loss terms. We present our overall method in Algorithm 1.

4 Experiments

4.1 Datasets and Evaluation Metrics

We use four multimodal datasets: UPMC-Food 101 [23], CMU-MOSI [27], IEMOCAP [3], and BraTS 2021 [1]. Table 1 presents the difference between these four datasets.

Table 1: The difference between the four datasets we use.

Dataset	Task type	No. of modality
UPMC-Food 101	Classification	2
CMU-MOSI	Regression	3
IEMOCAP	Classification	3
BraTS 2021	Segmentation	4

UPMC-Food 101 [23] is a food classification dataset, which contains about 100,000 recipes for a total of 101 food categories. Each item in the dataset is represented by one image plus textual information. We use accuracy and F1 score to evaluate the performance of the model.

CMU-MOSI [27] is a popular dataset for multimodal (audio, text and video) sentiment analysis. Each video segment is manually annotated with a sentiment score ranging from strongly negative to strongly positive (-3 to +3). Following previous work [18, 10], we use binary accuracy (ACC-2), F1 score, 7-class accuracy (ACC-7), mean absolute error (MAE) and pearson correlation (Corr) to evaluate the performance of the model.

Table 2: Quantitative results on the UPMC-Food 101 dataset. **Bold**: best results. Underline: second best results.

Method	Acc	F1
Text only	84.77	84.72
Image only	68.24	68.23
Baseline	90.32	90.30
G-Blending [22]	90.42	90.38
Greedy [25]	91.21	91.20
OGE [17]	91.08	91.08
AGM [15]	91.49	91.48
PMR [8]	92.01	91.98
UMT [7]	91.82	91.82
UME [7]	90.69	90.68
QMF [28]	92.87	92.85
ReconBoost [11]	92.52	92.51
CGGM	**92.94**	**92.90**

Table 3: Results on BraTS 2021. WT, TC and ET denote the dice score of Whole Tumor, Tumor Core and Enhancing Tumor respectively.

Method	WT	TC	ET	Avg.
flair only	70.42	51.41	45.27	55.70
t1 only	50.73	36.13	38.77	41.88
t2 only	64.82	42.17	34.62	47.20
t1ce only	56.61	53.83	53.14	54.53
Baseline	74.03	67.08	66.53	69.21
MRD	74.02	68.04	67.28	69.78
MSLR	74.47	69.98	67.20	70.55
UMT [7]	74.15	67.69	66.80	69.55
UME [7]	74.65	68.74	67.58	70.32
QMF [28]	75.11	70.78	68.94	71.61
ReconBoost [11]	75.21	70.18	70.01	71.80
CGGM	**76.94**	**72.75**	**72.14**	**73.94**

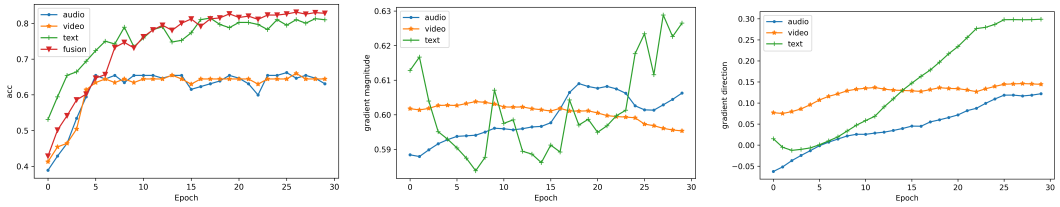
IEMOCAP [3] is a multimodal emotion recognition dataset, which contains recorded videos from ten actors in five dyadic conversation sessions. Following previous works [18, 24, 10], four emotions (happiness, anger, sadness and neutral state) are selected for emotion recognition. We use accuracy and F1 score to evaluate the performance of the model.

BraTS 2021 [1] is a 3D multimodal brain tumor segmentation dataset, which has four modalities: *flair, t1ce, t1* and *t2*. The input image size is $240 \times 240 \times 155$. The annotations are combined into three nested subregions: Whole Tumor (WT), Tumor Core (TC), and Enhancing Tumor (ET). We use Dice score of these three nested subregions and their average value to evaluate the performance.

https://doi.org/10.52202/079017-4238

Table 4: Quantitative results on the CMU-MOSI and IEMOCAP datasets. **Bold**: best results. Underline: second best results.

Method	CMU-MOSI					IEMOCAP	
	Acc-2	Acc-7	F1	MAE	Corr	Acc	F1
Text only	76.83	28.24	76.80	1.016	0.664	65.35	64.30
Audio only	64.12	23.04	66.96	1.451	0.510	52.18	50.14
Video only	62.00	21.65	66.18	1.441	0.499	54.55	53.97
Baseline	81.23	29.26	81.23	0.952	0.710	70.74	69.53
MRD	80.78	31.44	80.75	0.975	0.688	71.81	71.08
MSLR	81.22	31.00	81.19	0.950	0.704	71.99	70.96
AGM [15]	-	-	-	-	-	72.35	71.94
UMT [7]	81.78	31.98	81.77	0.942	0.712	70.75	69.81
UME [7]	80.83	30.94	80.78	0.969	0.701	71.53	70.94
QMF [28]	-	-	-	-	-	72.08	71.64
ReconBoost [11]	-	-	-	-	-	73.14	72.71
CGGM	**82.84**	**33.73**	**82.74**	**0.915**	**0.717**	**75.38**	**74.97**



(a) (b) (c)

Figure 3: Changes in (a) performance, (b) gradient magnitude and (c) direction during training with CGGM. We get the results on CMU-MOSI dataset.

4.2 Implementation Details

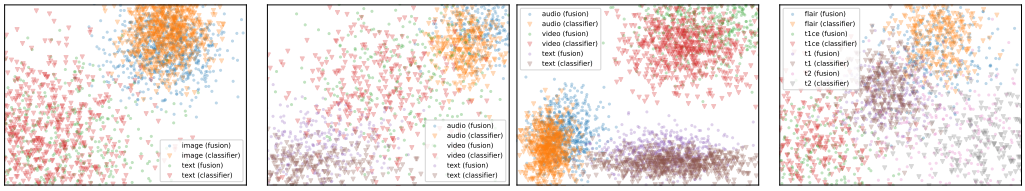
Input. For UPMC-Food 101, we use extracted features as inputs. Specifically, we use the pre-trained bert-base-uncased model [5] to extract text features and use pre-trained ViT [6] on ImageNet to extract image features. For CMU-MOSI and IEMOCAP, we follow Guo et al. [10] to extract acoustic, visual and textual features. For BraTS 2021, we use the preprocessed raw images as inputs.

Backbone. For UPMC-Food 101, CMU-MOSI and IEMOCAP, we use transformer encoders [20] as modality encoders and the fusion module. For the BraTS 2021 dataset, we use DeepLab v3+ [4] as the encoders and several convolution layers as the fusion module.

Training Details. For images in UPMC-Food 101 and BraTS 2021, we implement data augmentation strategies, including random cropping, random flipping, color jitter, adding noise, etc. To save memory, we consider BraTS 2021 as a 2D segmentation task by randomly slicing an image from the 3D image. For CMU-MOSI, we use L1 loss as our loss function. For UPMC-Food 101 and IEMOCAP, we use cross-entropy loss. For BraTS 2021, we use the combination of soft dice loss and cross-entropy loss. Besides, we use the Adam optimizer for CMU-MOSI, the AdamW optimizer for UPMC-Food 101 and IEMOCAP, and the SGD optimizer for BraTS 2021. Other hyperparameters are described in Appendix B in detail.

4.3 Main Results

Comparison with the state-of-the-arts. We compare our CGGM with other methods to demonstrate the effectiveness of our proposed method. For these four datasets, we compare CGGM with the model training only using one modality, multimodal joint training (Baseline), Modality Random Dropout (MRD), and Modality-specific Learning Rate (MSLR) methods. Additionally, we compare CGGM with SOTA methods including G-Blending [22], Greedy [25], OGM [17], AGM [15], PMR [8], UMT [7], UME [7], QMF [28] and ReconBoost [11]. Table 2, 3 and 4 present the results of



(a) UPMC-Food 101 (b) CMU-MOSI (c) IEMOCAP (d) BraTS 2021

Figure 4: t-SNE visualization of the gradients of classifiers and the unimodal gradients. Each point represents a gradient vector or matrix of a batch of data.

Table 5: Accuracy on IEMOCAP. f_1, f_2 and f_3 represent the audio, video and text classifier, respectively. We train three separate models in unimodal training.

unimodal training			multimodal training			CGGM		
audio	video	text	f_1	f_2	f_3	f_1	f_2	f_3
52.18	54.55	65.35	50.59	53.04	67.15	54.95	56.77	67.39

CGGM and its compared methods. Compared with the baseline method, CGGM brings significant improvements to the performance of the model, which demonstrates the effectiveness of our proposed method. Besides, CGGM takes both gradient magnitude and direction into consideration, thus making it outperform other gradient modulation methods consistently in all four datasets. Most importantly, CGGM can be easily applied to various tasks and has good performance, including classification tasks, regression tasks, segmentation tasks, etc. Meanwhile, CGGM has no limitations on the optimizer, loss function and the number of modalities.

Effectiveness of CGGM. In Figure 3, we visualize the changes in accuracy, gradient magnitude and direction during training with CGGM. Compared with Figure 2(a), the accuracy of text modality in Figure 3(a) does not increase very fast with CGGM, which indicates that CGGM imposes constraints to the dominant modality during the optimization process. Besides, the accuracies of all the modalities and the fusion improves, indicating the effectiveness of CGGM. Additionally, in Figure 2(b), the dominant modality always has the largest gradient while in Figure 3(b), the gradient magnitude of the text modality decreases at first, indicating that CGGM slows down its optimization and accelerates other modalities' optimization, helping each modality learn sufficiently, thus improving the multimodal performance. In case of gradient direction, in Figure 2, the similarity between audio modality and the fusion is always less than 0 during the training process, indicating an opposite optimization direction between the unimodal and multimodal, thus hindering the optimization process. In Figure 3, we observe the multimodal direction is consistent with all modalities, indicating that the multimodal branch utilizes unimodal information efficiently.

4.4 Classifier Performance and Gradient Direction

Classifier performance. In Table 5, we present the accuracy of classifiers in different situations. Unimodal training can be considered a baseline that fully utilizes the unimodal information. Compared with unimodal training, the accuracies of f_1 and f_2 in multimodal training drop slightly while the accuracy of f_3 increases slightly. This demonstrates that multimodal training can not fully utilize the information from audio and video, indicating that they are under-optimized. The improvement of f_3 indicates that the text modality is fully exploited and learns some information from the other two modalities. In contrast, the accuracies of the three classifiers in CGGM all improve. This suggests that during the balancing process, the fusion can fully utilize the information from all the modalities, thus in turn making the encoders of three modalities fuse the information from other modalities during backpropagation. Therefore, the accuracies of all the three classifiers improve correspondingly. This also validates the effectiveness of CGGM.

Classifier gradient direction. In Section 3.3.2, we propose to use the gradients of classifiers to represent the unimodal gradients. In this subsection, we give a visualization of the gradients of classifiers and the unimodal gradients. Specifically, for every batch of data, we input them into

https://doi.org/10.52202/079017-4238

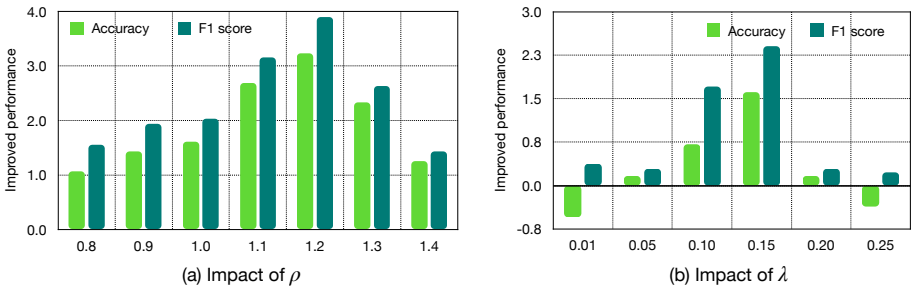


Figure 5: The improved performance with different ρ and λ compared to the joint training baseline.

the model to get representations h_i which are then fed into the classifiers f_i to get the gradients of classifiers. Then we input h_i of only one modality into the fusion module Ω to get the unimodal gradients. We use t-SNE [19] to visualize the gradient vectors. Figure 4 shows the visualization results on the four datasets. As shown in the figure, for each modality, the unimodal gradient vectors and the gradient vectors of the corresponding classifier are very close to each other, demonstrating that it is meaningful to use the gradients of the classifiers to represent the unimodal gradients.

4.5 Ablation Study

Gradient magnitude and direction. To measure the contribution of gradient magnitude modulation and gradient direction modulation separately, we present our ablation results on IEMOCAP in Table 6. Compared with the baseline in the first row, modulating the magnitude of the gradients brings more improvements to the performance of the model than modulating the direction of the gradients. Compared with the CGGM performance in Table 4, the combination of modulating gradient magnitude and gradient directions furthermore enhances the performance of the model.

Scaling hyperparameter ρ. To explore the impacts of the scaling hyperparameter ρ on the model's performance, we select seven different values of ρ and present our results on IEMOCAP in Figure 5(a). We discover that the accuracy improves with the increase of ρ before hitting the highest value when $\rho = 1.2$. Then, the accuracy drops with the increase of ρ. Compared to the baseline, modulating the magnitude of the gradients brings consistent improvements regardless of how big ρ is taken. Intuitively, the larger the ρ, the larger the modification to the gradients. Therefore, the results in the table indicate that we need to carefully choose a ρ to avoid modifications that are too large or too small.

Table 6: The benefits of modulating the magnitude of the gradients and the directions of the gradients.

Model	Acc	F1
Baseline	70.74	69.53
CGGM ($\rho = 1.0, \lambda = 0$)	72.35	71.56
CGGM ($\rho =$ None, $\lambda = 0.1$)	72.41	72.07
CGGM ($\rho = 1.0, \lambda = 0.1$)	73.74	73.18

Loss trade-off λ. λ measures the strength we modulate the directions of the gradients. We select six different values of λ and present the results on IEMOCAP in Figure 5(b). As shown in the figure, when λ is 0.01 or 0.25, the accuracy will decrease slightly. When λ is too small, the modulation is insufficient and could influence the optimization process. When λ is too large, the modulation is large and will influence the task loss, thus making optimization deviate from the task.

5 Conclusion

In this paper, we propose CGGM, a novel strategy to balance the multimodal training process. Compared to existing gradient modulation methods, CGGM has no limitations on the loss functions, the optimizer, the number of modalities, etc. Moreover, we consider both the magnitude and direction of the gradients with the guidance of the classifiers. Extensive experiments and ablation studies fully demonstrate the effectiveness and universality of CGGM. However, CGGM also has a limitation. CGGM needs to leverage extra classifiers to implement gradient modulation. Although these classifiers are lightweight, they still lead to more computational resources. We lead this challenging problem to future work.

Acknowledgement

This work was supported by National Key R&D Program of China under Grant No.2022ZD0162000.

References

[1] Ujjwal Baid, Satyam Ghodasara, Suyash Mohan, Michel Bilello, Evan Calabrese, Errol Colak, Keyvan Farahani, Jayashree Kalpathy-Cramer, Felipe C Kitamura, Sarthak Pati, et al. The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. *arXiv preprint arXiv:2107.02314*, 2021.

[2] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2612–2620, 2017.

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

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Chenzhuang Du, Jiaye Teng, Tingle Li, Yichen Liu, Tianyuan Yuan, Yue Wang, Yang Yuan, and Hang Zhao. On uni-modal feature learning in supervised multi-modal learning. In *International Conference on Machine Learning*, pages 8632–8656. PMLR, 2023.

[8] Yunfeng Fan, Wenchao Xu, Haozhao Wang, Junxiao Wang, and Song Guo. Pmr: Prototypical modal rebalance for multimodal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20029–20038, 2023.

[9] Jie Fu, Junyu Gao, Bing-Kun Bao, and Changsheng Xu. Multimodal imbalance-aware gradient modulation for weakly-supervised audio-visual video parsing. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[10] Zirun Guo, Tao Jin, and Zhou Zhao. Multimodal prompt learning with missing modalities for sentiment analysis and emotion recognition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1726–1736, 2024.

[11] Cong Hua, Qianqian Xu, Shilong Bao, Zhiyong Yang, and Qingming Huang. Reconboost: Boosting can achieve modality reconcilement. In *The Forty-first International Conference on Machine Learning*, 2024.

[12] Yu Huang, Junyang Lin, Chang Zhou, Hongxia Yang, and Longbo Huang. Modality competition: What makes joint training of multi-modal network fail in deep learning?(provably). In *International Conference on Machine Learning*, pages 9226–9259. PMLR, 2022.

[13] Tao Jin, Weicai Yan, Ye Wang, Sihang Cai, Shuaiqifan, and Zhou Zhao. Calibrating prompt from history for continual vision-language retrieval and grounding. In *ACM Multimedia 2024*, 2024.

 https://doi.org/10.52202/079017-4238

[14] Hamid Reza Vaezi Joze, Amirreza Shaban, Michael L Iuzzolino, and Kazuhito Koishida. Mmtm: Multimodal transfer module for cnn fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13289–13299, 2020.

[15] Hong Li, Xingyu Li, Pengbo Hu, Yinuo Lei, Chunxiao Li, and Yi Zhou. Boosting multi-modal model performance with adaptive gradient modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22214–22224, 2023.

[16] Paul Pu Liang, Ziyin Liu, AmirAli Bagher Zadeh, and Louis-Philippe Morency. Multimodal language analysis with recurrent multistage fusion. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 150–161, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1014. URL https://aclanthology.org/D18-1014.

[17] Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8238–8247, 2022.

[18] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6558–6569, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1656. URL https://aclanthology.org/P19-1656.

[19] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[21] Valentin Vielzeuf, Alexis Lechervy, Stéphane Pateux, and Frédéric Jurie. Centralnet: a multilayer approach for multimodal fusion. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[22] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12695–12705, 2020.

[23] Xin Wang, Devinder Kumar, Nicolas Thome, Matthieu Cord, and Frédéric Precioso. Recipe recognition with large multimodal food dataset. In *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–6, 2015. doi: 10.1109/ICMEW.2015.7169757.

[24] Yansen Wang, Ying Shen, Zhun Liu, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7216–7223, 2019.

[25] Nan Wu, Stanislaw Jastrzebski, Kyunghyun Cho, and Krzysztof J Geras. Characterizing and overcoming the greedy nature of learning in multi-modal deep neural networks. In *International Conference on Machine Learning*, pages 24043–24055. PMLR, 2022.

[26] Weicai Yan, Ye Wang, Wang Lin, Zirun Guo, Zhou Zhao, and Tao Jin. Low-rank prompt interaction for continual vision-language retrieval. In *ACM Multimedia 2024*, 2024.

[27] Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems*, 31 (6):82–88, 2016. doi: 10.1109/MIS.2016.94.

[28] Qingyang Zhang, Haitao Wu, Changqing Zhang, Qinghua Hu, Huazhu Fu, Joey Tianyi Zhou, and Xi Peng. Provable dynamic fusion for low-quality multimodal data. In *International conference on machine learning*, pages 41753–41769. PMLR, 2023.

A Gradient Direction Modulation Details

For classification tasks, the classifier f_i outputs more than one value. For example, for the UPMC-Food 101 dataset, f_i output 101 values for each piece of data. Therefore, we can define f_i as $f_i = (f_i^{(1)}, f_i^{(2)}, \cdots, f_i^{(m)})$ where m is the number of output. We calculate the gradients of the classifiers f_i as:

$$\nabla_{\theta^{f_i}} \mathcal{L}(\theta^{f_i}) = \frac{\partial \mathcal{L}(\theta^{f_i})}{\partial f_i} = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta^{f_i^{(1)}})}{\partial \theta_1^{f_i}} & \frac{\partial \mathcal{L}(\theta^{f_i^{(2)}})}{\partial \theta_1^{f_i}} & \cdots & \frac{\partial \mathcal{L}(\theta^{f_i^{(m)}})}{\partial \theta_1^{f_i}} \\ \frac{\partial \mathcal{L}(\theta^{f_i^{(1)}})}{\partial \theta_2^{f_i}} & \frac{\partial \mathcal{L}(\theta^{f_i^{(2)}})}{\partial \theta_2^{f_i}} & \cdots & \frac{\partial \mathcal{L}(\theta^{f_i^{(m)}})}{\partial \theta_2^{f_i}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \mathcal{L}(\theta^{f_i^{(1)}})}{\partial \theta_n^{f_i}} & \frac{\partial \mathcal{L}(\theta^{f_i^{(2)}})}{\partial \theta_n^{f_i}} & \cdots & \frac{\partial \mathcal{L}(\theta^{f_i^{(m)}})}{\partial \theta_n^{f_i}} \end{bmatrix} \tag{14}$$

Similarly, the gradients of the fusion module classifier can be calculated as:

$$\nabla_{\theta^{\mathcal{F}}} \mathcal{L}(\theta^{\mathcal{F}}) = \frac{\partial \mathcal{L}(\theta^{\mathcal{F}})}{\partial \mathcal{F}} = \begin{bmatrix} \frac{\partial \mathcal{L}(\theta^{\mathcal{F}^{(1)}})}{\partial \theta_1^{\mathcal{F}}} & \frac{\partial \mathcal{L}(\theta^{\mathcal{F}^{(2)}})}{\partial \theta_1^{\mathcal{F}}} & \cdots & \frac{\partial \mathcal{L}(\theta^{\mathcal{F}^{(m)}})}{\partial \theta_1^{\mathcal{F}}} \\ \frac{\partial \mathcal{L}(\theta^{\mathcal{F}^{(1)}})}{\partial \theta_2^{\mathcal{F}}} & \frac{\partial \mathcal{L}(\theta^{\mathcal{F}^{(2)}})}{\partial \theta_2^{\mathcal{F}}} & \cdots & \frac{\partial \mathcal{L}(\theta^{\mathcal{F}^{(m)}})}{\partial \theta_2^{\mathcal{F}}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \mathcal{L}(\theta^{\mathcal{F}^{(1)}})}{\partial \theta_n^{\mathcal{F}}} & \frac{\partial \mathcal{L}(\theta^{\mathcal{F}^{(2)}})}{\partial \theta_n^{\mathcal{F}}} & \cdots & \frac{\partial \mathcal{L}(\theta^{\mathcal{F}^{(m)}})}{\partial \theta_n^{\mathcal{F}}} \end{bmatrix} \tag{15}$$

In order to calculate cosine similarity between these two terms, we flatten them into vectors and then use Equation 12 to calculate $\mathcal{L}_{gm}$.

B Implementation Details

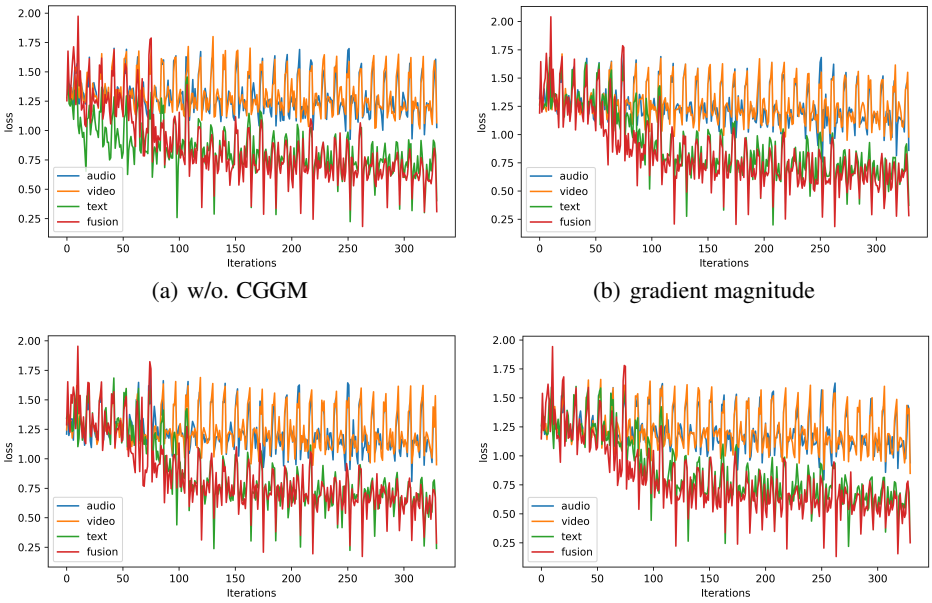
Table 7: Main hyperparameters of the four datasets.

Hyperparameters	UMPC-Food 101	CMU-MOSI	IEMOCAP	BraTS 2021
batch size	128	64	32	64
optimizer	AdamW	Adam	AdamW	SGD
base lr	1e-3	1e-3	1e-3	1e-2
classifier lr	5e-4	5e-4	5e-4	6e-3
weight decay	3e-3	-	-	3e-4
gradient clip	0.8	0.8	0.8	-
scheduler	StepLR	StepLR	StepLR	CosineLR
ρ	1.3	1.3	1.3	1.2
λ	0.15	0.20	0.15	0.10
warm-up epoch	-	-	-	10
epoch	80	30	30	150

Table 7 presents the main hyperparameters of the four datasets. Apart from the hyperparameters in the table, there are some task-specific hyperparameters.

For BraTS 2021, the start learning rate is set to 4e-4 with warm-up epochs to 1e-2 and the final learning rate is 1e-3. Besides, for the loss function, we use the combination of soft dice loss and cross-entropy loss, which can be represented as $\mathcal{L}_{task} = \mathcal{L}_{Dice} + \lambda_1 \mathcal{L}_{CE}$. We set λ_1 to 1. Particularly, we use a weighted cross-entropy loss function, where the weight is 0.2, 0.3, 0.25 and 0.25 for the background, label 1, label 2 and label 3, respectively.

https://doi.org/10.52202/079017-4238



(a) w/o. CGGM (b) gradient magnitude

(c) gradient direction (d) CGGM

Figure 6: Changes in loss during the training process.

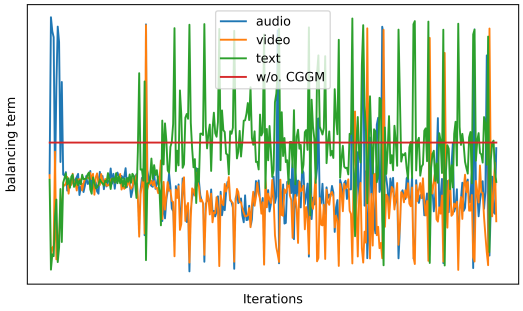


Figure 7: Changes in balancing term during the training process.

C More Ablation Study

More visualizations of CGGM. We further visualize the loss changes in Figure 6. From the figure, we can observe that the loss of the dominant modality with CGGM implemented in (b)-(d) will drop much slower than that in Figure 6(a). Besides, the losses of all modalities in (b)-(d) are smaller than those in (a), indicating the effectiveness of CGGM. Apart from the loss changes, we also visualize the changes in balancing term during the training process in Figure 7. When the value is higher than the red line, the modality is promoted. When the value is lower than the red line, the modality is suppressed. In the first few iterations, the dominant modality is suppressed, ensuring that other modalities are fully optimized. During the optimization, balancing terms of three modalities turn up and down, ensuring each modality is sufficiently optimized.

Additional computational resources of classifiers. The additional classifiers will need more computational resources during training. However, during inference, the classifiers will be discarded. Therefore, they have no impact during the inference stage. We report the additional memory cost (MB) of the additional classifiers in Table 8. From the table, we can observe that the additional

https://doi.org/10.52202/079017-4238

Table 8: Additional gpu memory cost (MB) of classifiers.

Setting	Food101	MOSI	IEMOCAP	BraTS
With classifiers	+8MB	+8MB	+8MB	+24MB

computational increase is low. There are two main reasons: (1) the classifiers or decoders are light with only a few parameters; (2) the classifiers only use the gradients to update themselves and do not pass the gradients to the modality encoders during backpropagation. Therefore, there is no need to store the gradient for each parameter, thus reducing memory cost.

https://doi.org/10.52202/079017-4238

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Abstract and Introduction

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Conclusion Section

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The Method Section

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We present the detailed algorithm, codes and hyperparameters.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have released our codes.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the detailed information in the implementation details.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: [NA]

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The experiments are not computationally expensive.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: [NA]

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: [NA]

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: [NA]

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: [NA]

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: [NA]

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

https://doi.org/10.52202/079017-4238