Algorithm 1: The pseudo code of in DI-MML.

Input: Input data $\mathcal{D} = \left\{x_i^1, x_i^2, y_i\right\}_{i=1,2,...,N}$, initialized encoders ϕ^1, ϕ^2, classifiers ψ^1, ψ^2, and ψ^s, and fusion module ψ^f. Hyper-parameters λ_s, λ_D, epoch number E, warmup epoch number E_w, fusion epoch E_f.

```
int e=0;
Encoder training:
while e < E do
    if e = E_m then
        Calculate the dimension-wise prediction using Eq. 4,
        obtain effective and ineffective dimensions using Eq.
         5 (can perform only once)
    end
    foreach mini-batch data B_t in D at step t do
        if e < E_m then
            Calculate the loss L^i = L^i_CE + λ_s L^Si_CE
        else
            Calculate the final loss L^i with Eq. 7
        end
        Update networks φ^1, φ^2, ψ^1, ψ^2, ψ^s for different
         modalities with L^i.
    end
    e=e+1;
end
Fusion module training::
while e < E_f do
    Freeze φ^1, φ^2 and update ψ^f according to L^f_CE.
end
```

A BASELINES

In this paper, we compare our method with eight multimodal baselines and we give the description of them below.

Joint training: Joint training is the most basic multimodal training framework with concatenation fusion on the extracted features from different modalities and then input into a linear classifier while the network is trained with the cross-entropy loss.

MSES: Modality-Specific Early Stop (MSES) [9] restrain the decrease in overall accuracy of the model by detecting the occurrence of overfitting in each modality and individually controlling the learning process. The detected overfitted modality will be stopped first.

MSLR: Modality-Specific Learning Rate (MSLR) [41] uses different learning rates for different modalities while training an additive late-fusion model. It contains "Keep", "Smooth" and "Dynamic" strategies and in this paper we compare with its "Dynamic" strategy because of its better performance.

OGM-GE: On-the-fly Gradient Modulation (OGM-GE) [22] dynamically controls the optimization of each modality based on their contribution to the learning objective. By monitoring and adapting the gradients, the method aims to address the imbalance problem without the need for additional neural modules.

PMR: Prototypical Modal Rebalance (PMR) [8] focuses on stimulating the slow-learning modality without interference from other modalities. Using prototypes could help to regulate the learning directions and paces of modality-specific gradient.

UMT: Unimodal Teacher (UMT) [6] distills the pre-trained uni-modal features to the corresponding parts in multi-modal networks in multi-modal training. Uni-modal distillation happens before fusion, so it's suitable for late-fusion multi-modal architecture. The pre-trained uni-modal features are generated by inputting the data to the pre-trained uni-modal models.

MM Clf and Preds Avg: They are as described in Section 3.1.

B TRAINING SCHEME

The details of training scheme is shown here as well as the pseudo code. The randomly initialized neural networks perform worse and cannot be used to identify the informative dimensions. Therefore, we perform unimodal training independently with unimodal cross-entropy loss for some warmup epochs (10 in our experiments). And then, the shared classifier and DUC loss are applied for the left encoder training epochs. After encoder training, we fix the parameters of encoders and train a linear fusion classifier by concating multimodal features.