Prerequisites

To understand the paper completely we should a basic understanding of following topics:

1. LSTM and Bi-LSTM
2. BERT model
3. Fusion networks like:
   - Tensor Fusion Network (TFN)
   - Low Rank Multimodal Fusion (LMF)
   - Memory Fusion Network
   - Graph Fusion Network
4. Domain Separation Network (DSN)
5. Adversial Technique and Adversial loss

Introduction

Firstly, in order to effectively utilize redundant information, the joint domain separation representations of all modes are obtained through the improved joint domain separation network. Then, the hierarchical graph fusion net (HGFN) is used for dynamically fusing each representation to obtain the interaction of multimodal data for guidance in the sentiment analysis.

Multimodal Data and Feature Dimensions:

- Multimodal data includes sequences from different modalities (e.g., speech, video, and text), which often have **different feature dimensions** and **temporal dependencies**.

- Processing such data requires models that can handle these variations and learn meaningful representations.

Long Short-Term Memory (LSTM):

- LSTM is a type of recurrent neural network (RNN) designed to process sequential data.

- It captures long-term dependencies and temporal patterns, making it a powerful tool for extracting features from each modality.

Challenges with Single LSTM Models:

- Using a single LSTM for multimodal tasks can be insufficient because it may fail to capture **shared features** (common information across modalities) and **specialized features** (unique aspects of each modality).

- Redundancy and incomplete utilization of cross-modal information are common issues.

Advanced Multimodal Models: (Based on different kinds of fusion)

1. **Memory Fusion Network (MFN)** and **Graph-MFN**:
   - These methods extend LSTMs to handle multimodal data by incorporating **memory structures** to track temporal and cross-modal relationships.

2. **Tensor Fusion Network (TFN)**:

    - Represents different modalities as tensors and learns their interactions during the fusion process.
    - However, it does not fully utilize inter-modal dependencies before fusion.

3. **Low-Rank Multimodal Fusion (LMF)**:
    - Reduces the computational complexity of multimodal fusion while capturing cross-modal interactions.
    - Like TFN, it also lacks mechanisms to fully exploit inter-modal relationships before fusion.

Domain Separation Network (DSN)

A **Domain Separation Network (DSN)** is a machine learning method designed to handle data from multiple sources (like audio, text, or video). Here's how it works in simpler terms:

Core Idea:

DSN breaks down the features (characteristics) of the data into two types:

1. **Shared Features**: Information that is common across all data sources (e.g., tone of speech might relate to emotion in both audio and video).
2. **Specialized Features**: Information unique to each data source (e.g., body language is specific to video, while pitch is specific to audio).

In this paper DSN was improved to make JDSN.

What Is Improved JDSN?

Improved JDSN is used to learn two kinds of representations from multimodal data:

1. **Modality-Invariant Representations** (Shared Features):
    - These are common features across all data types (e.g., emotional tone found in both speech and facial expressions).
    - The goal is to map all modalities into a shared space where their features are closely aligned, reducing differences between modalities.
    - This alignment makes it easier to fuse the information from different modalities effectively.
2. **Modality-Specific Representations** (Specialized Features):
    - These are unique features specific to each data type (e.g., facial expressions in video or pitch in audio).
    - These specialized features act as additional complementary information that enriches the overall understanding.

Why Combine These Representations?

- By combining the **shared features** (common understanding) and **specialized features** (unique aspects), the model can make full use of all available information, leading to better predictions.

Now these features are fed to a fusion network called HGFN.

What is HGFN?

The **Hierarchical Graph Fusion Network (HGFN)** improves upon these early methods by introducing a **dynamic fusion process**. Here's how it works:

1. **Dynamic Layers**:
    - **Unimodal Layer**: Focuses on the unique features of each individual modality (e.g., only text, only audio).
    - **Bimodal Layer**: Examines how two modalities interact with each other (e.g., how audio and video together convey emotion).
    - **Trimodal Layer**: Considers all three modalities together (text, audio, video) to capture higher-level interactions.
2. **Interactive Learning Process**:
    - The outputs from these layers are connected to create a hierarchical structure where the model learns progressively richer interactions between modalities.
    - This process adjusts dynamically, assigning different weights to each modality depending on how relevant it is for the task.

Combining Improved JDSN and HGFN

While HGFN improves multimodal fusion, it still struggles with **redundant information** (repeated or unnecessary details across modalities). The **Improved JDSN** addresses this by:

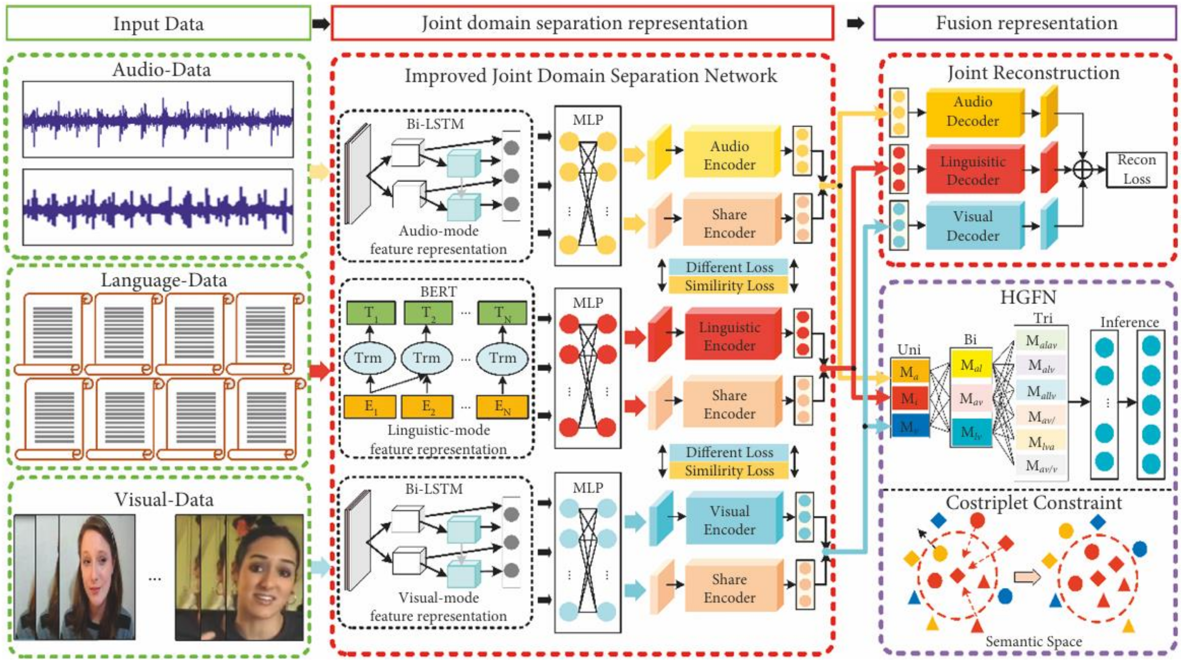
- **Reducing redundancy**: Ensuring that shared and specific features are cleanly separated.
- **Enhancing modal interactions**: Using these features to improve how modalities interact during fusion.

Advantages of combining JDSN and HGFN

By combining Improved JDSN and HGFN:

1. **Efficient Representation**:
    - The Improved JDSN aligns and separates modality-specific and shared features, reducing the complexity of the fusion process.
2. **Dynamic Fusion**:
    - The HGFN enables the model to dynamically adjust how much each modality contributes to the final prediction.

3. **Maximized Information Use**:
    - Together, these methods capture richer interactions and make better use of redundant information, resulting in a more robust and accurate multimodal model.



In summary:

Improvements to DSN:

1. **Extended Mode**: Enhanced to handle more modalities.
2. **Orthogonal Constraint Loss**:

    This introduces a new loss function to ensure that **specialized features** (unique to each modality) from different data sources remain distinct and do not overlap.

    This helps the model disentangle information more effectively, reducing redundancy.
3. **Advanced Similarity Metric (CMD)**: Replaces adversarial loss for better feature alignment.

    The **adversarial loss** previously used to align features across modalities is replaced with **Central Moment Discrepancy (CMD)**.

    CMD is a more precise and robust metric for measuring similarity between feature distributions of different modalities, improving alignment.
4. **Joint Representation**: Combines shared and specific features at the output for richer information.

DISRFN:

- **Improved JDSN**: Aligns shared features and extracts unique ones while reducing redundancy.
- **HGFN**: Dynamically fuses interactions at unimodal, bimodal, and trimodal levels.

In multimodal sentiment analysis, the mainstream multi modal learning methods include **multimodal fusion representation** and **multimodal representation learning.**

1. Multimodal Fusion Representation: HGFN
2. Multimodal Representation Learning:

   focuses on representing data from multiple modalities (e.g., text, audio, video) in a unified way. It divides approaches into two main types: **common subspace representations** and **factorized representations.**

# Common Subspace Representation

These methods aim to map data from different modalities into a shared space where relationships between them can be analysed. They are divided into two types:

(a) Correlation-Based Models:

- Focus on finding correlations between modalities to learn shared representations.

(b) Adversarial Learning-Based Models:

- Use adversarial techniques to enforce cross-modal invariance (i.e., making shared representations independent of specific modality characteristics). Example: GANs

Limitations: These models mainly focus on **shared representations** and neglect the **specialized features** unique to each modality.

# Factorized Representations

These methods **separate data** into shared and modality-specific components, typically using matrix factorization techniques.

Limitations: Matrix decomposition often results in **incomplete feature representations**, leading to loss of useful information.

The improved JDSN in this paper overcomes the limitations of the above methods.

The framework of DISRFN is shown in Figure 1:

(1) The data of the three modes are fed into the corresponding Bi-LSTM and BERT models to obtain the discourse-level feature representations;

(2) The discourse-level feature representations of each mode are fed into the corresponding MLP (multilayer perception) to obtain the representation of unified dimension;

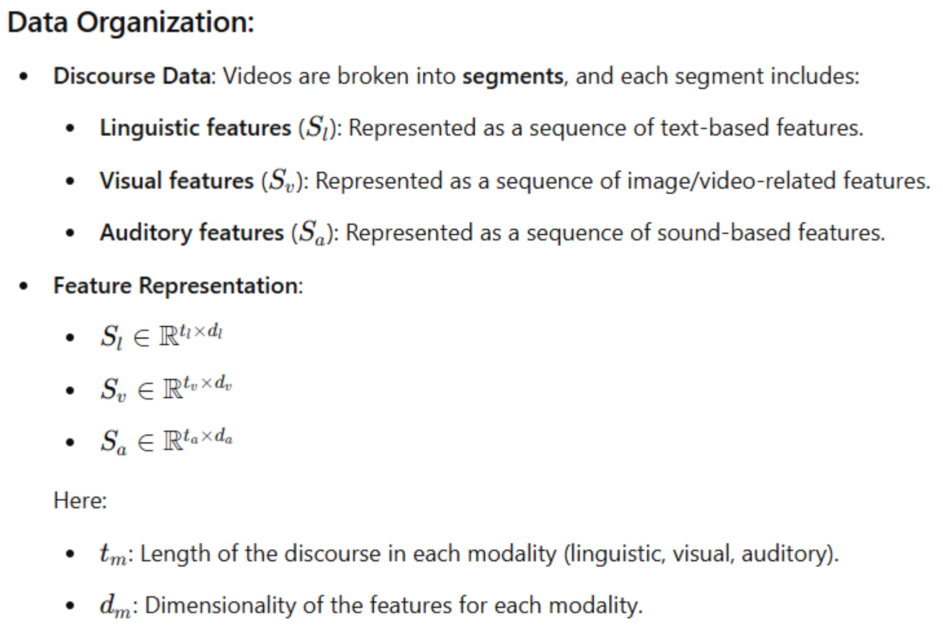
(3) The unified representations of each mode are fed into the corresponding encoder and shared encoder to obtain the shared representations and special representations;

(4) The shared representations are added with a special representation of each modal to obtain the joint domain separation representations;

(5) The joint domain separation representations of each mode are fed into the corresponding decoder to obtain the reconstruction loss;

(6) The joint domain separation representations of each mode are fed into HGFN for dynamic fusion to perform MSA task

DISFRN Model



Data Organization:

- **Discourse Data**: Videos are broken into **segments**, and each segment includes:
  - **Linguistic features** (S_l): Represented as a sequence of text-based features.
  - **Visual features** (S_v): Represented as a sequence of image/video-related features.
  - **Auditory features** (S_a): Represented as a sequence of sound-based features.
- **Feature Representation**:
  - $S_l \in \mathbb{R}^{t_l \times d_l}$
  - $S_v \in \mathbb{R}^{t_v \times d_v}$
  - $S_a \in \mathbb{R}^{t_a \times d_a}$

  Here:

  - t_m: Length of the discourse in each modality (linguistic, visual, auditory).
  - d_m: Dimensionality of the features for each modality.

Read section 3.2 from paper

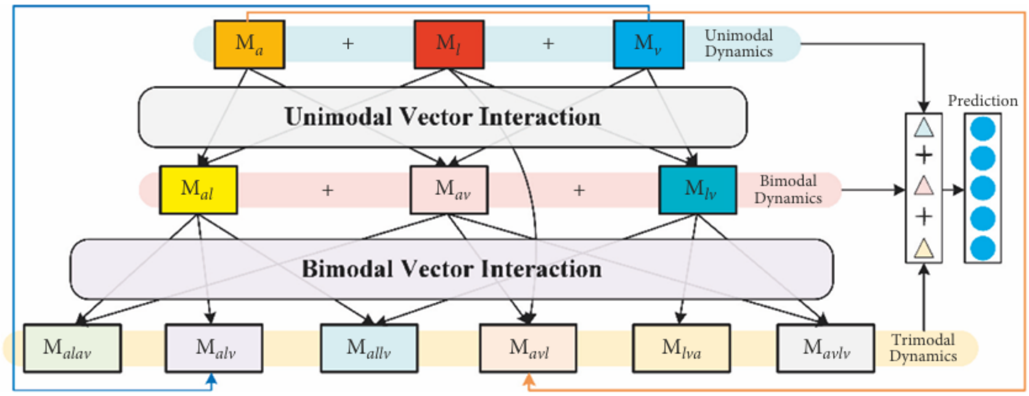


FIGURE 2: The framework of HGFN.

Loss Function

A joint loss function is newly set to effectively learn the network model:

$$\boxed{L_{total} = L_{task} + \alpha L_{diff} + \beta L_{sim} + \gamma L_{recon} + \eta L_{trip}}$$

1. Task Loss

The mean square error (MSE) is used as the task loss of the network to predict continuous dense variable.

$$L_{\text{task}} = \frac{1}{N_b}\sum_{i=0}^{N_b}\left\|y_i - \hat{y}_i\right\|_2^2.$$

N_b = discourse data in one batch;

Y_i refers to the actual emotional label;

$Y_i^{\wedge}$ refers to the predictive value of the network;

2. Differential Loss

$$L_{\text{diff}} = \sum_{m\in\{l,v,a\}} \left\|H_m^{c\top} H_m^{p}\right\|_F^2 + \sum_{\substack{(m_1,m_2)\in\{(l,a),\\(l,v),(a,v)\}}} \left\|H_{m_1}^{p\top} H_{m_2}^{p}\right\|_F^2,$$

where $\|\cdot\|_F^2$ refers to squared Frobenius norm.

The loss function L_{diff} has two main terms:

(i). Intra-Modal Orthogonality:

$$\sum_{m\in\{l,v,a\}} \left\|H_m^{c\top} H_m^{p}\right\|_F^2$$

- This term ensures that within each modality *(l = language, v= vision, a = audio)*, the **modality-invariant representations ($H_m{}^c$)** and **modality-specific representations ($H_m{}^p$)** are orthogonal to each other.

- The Frobenius norm enforces a penalty if these representations are not independent. This separation ensures that shared features (invariant) are disentangled from unique features (specific) within the same modality.

(ii). Inter-Modal Orthogonality:

$$\sum_{(m_1,m_2)\in\{(l,a),(l,v),(a,v)\}} \left\|H_{m_1}^{p\top} H_{m_2}^{p}\right\|_F^2$$

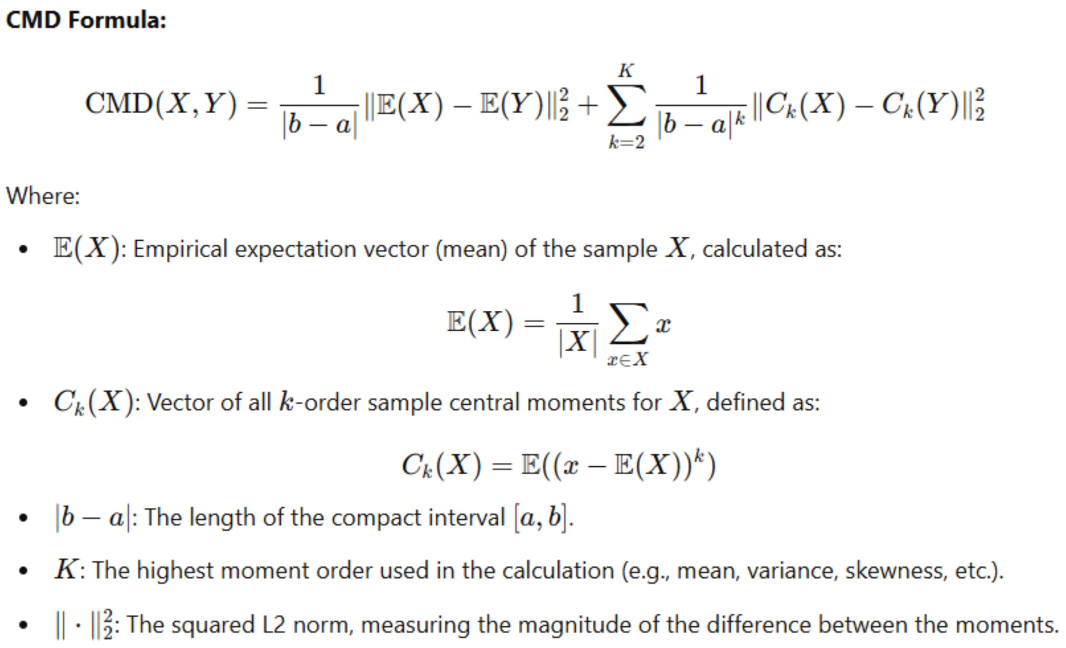
- This term ensures that **modality-specific representations** across different modalities are orthogonal. For example, audio-specific features **($H_a{}^p$)** should be independent of language-specific **($H_l{}^p$)** or vision-specific **($H_v{}^p$)** features.

- This enforces diversity among the specific representations across modalities

3. Similarity Loss

CMD (Central Moment Discrepancy)

The **Central Moment Discrepancy (CMD)** is a metric used to measure the difference between two probability distributions p and q. CMD evaluates the discrepancy by comparing the **moments** (statistical properties like mean, variance, skewness, etc.) of two random samples X and Y, ensuring their distributions align over a compact interval $[a,b]^N$.



CMD Formula:

$$\mathrm{CMD}(X,Y) = \frac{1}{|b-a|}\|\mathbb{E}(X)-\mathbb{E}(Y)\|_2^2 + \sum_{k=2}^{K}\frac{1}{|b-a|^k}\|C_k(X)-C_k(Y)\|_2^2$$

Where:

- $\mathbb{E}(X)$: Empirical expectation vector (mean) of the sample X, calculated as:

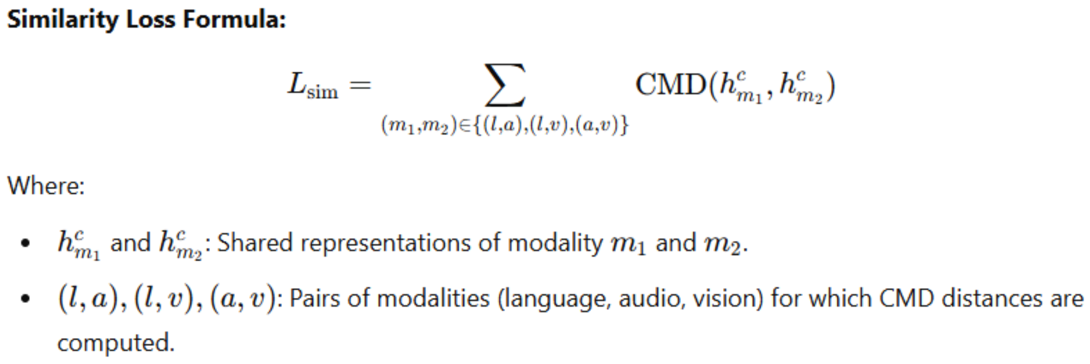
$$\mathbb{E}(X) = \frac{1}{|X|}\sum_{x\in X} x$$

- $C_k(X)$: Vector of all k-order sample central moments for X, defined as:

$$C_k(X) = \mathbb{E}((x-\mathbb{E}(X))^k)$$

- $|b-a|$: The length of the compact interval $[a,b]$.
- K: The highest moment order used in the calculation (e.g., mean, variance, skewness, etc.).
- $\|\cdot\|_2^2$: The squared L2 norm, measuring the magnitude of the difference between the moments.

The **Similarity Loss** is designed to **reduce the heterogeneity** (divergence) between the shared representations of different modalities *(l = language, v= vision, a = audio)*. This helps align the shared subspaces of these modalities, enabling better multimodal integration.



Similarity Loss Formula:

$$L_{\mathrm{sim}} = \sum_{(m_1,m_2)\in\{(l,a),(l,v),(a,v)\}} \mathrm{CMD}(h_{m_1}^c, h_{m_2}^c)$$

Where:

- $h_{m_1}^c$ and $h_{m_2}^c$: Shared representations of modality m_1 and m_2.
- (l,a), (l,v), (a,v): Pairs of modalities (language, audio, vision) for which CMD distances are computed.

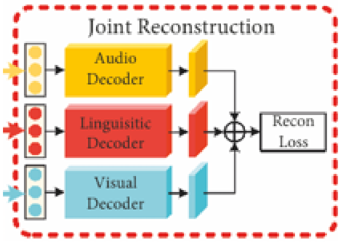
By minimizing L_{sim} , the network ensures that shared features across modalities (e.g., audio and video) are consistent and can be effectively fused for tasks like sentiment or emotion analysis.

For example:

If audio says "cheerful tone," language should not contradict it by saying "negative sentiment."

4. Reconstruction Loss

Reconstruction loss ensures that the encoder captures the **detailed and meaningful representations** for each modality by reconstructing the input back from the encoded representations.



- **Formula**:

$$L_{\text{recon}} = \frac{1}{3} \sum_{m \in \{l,v,a\}} \|h_m - \hat{h}_m\|_2^2$$

where:

- h_m: Original joint representation vector for modality m (language, vision, audio).
- $\hat{h}_m$: Reconstructed representation vector for the same modality, produced by the decoder D_m.
- $\|\cdot\|_2^2$: Squared L2 norm, which measures the error between h_m and $\hat{h}_m$.

Why Reconstruction Loss is Needed:

Without **Reconstruction Loss** , the following **problems can arise**:

1. **Trivial Representations in Specific Encoders**:
   - Specific encoders (E_m^p) may end up learning representations that lack meaningful details about the modality.
2. **Underfitting of Encoders**: Encoders may focus only on separating features but fail to retain input-specific details.
3. **Balancing Detail and Abstraction**: Ensures encoders preserve enough information for meaningful reconstruction.

How L_{recon} Solves the Problems:

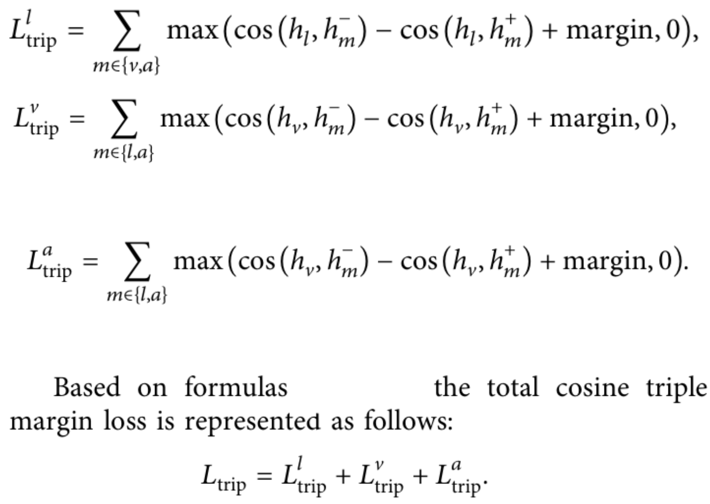
1. **Ensures Information Preservation**:
   - The reconstruction loss penalizes the model if the decoder D_m cannot reconstruct the original input h_m from the encoded representation $h^{\wedge}_{m}$.
   - This forces the encoder to **retain all relevant details** about the modality, even while separating shared and specific features.
2. **Avoids Trivial Representations**: Penalizes oversimplified patterns by requiring rich encoded details.
3. **Improves Generalization**: Creates detailed representations suited for downstream tasks like sentiment or emotion detection.

5. Cosine Triplet Margin Loss

Why Triplet Margin Loss is Needed:

- **Aligns Similar Representations Across Modalities**:
  1. Ensures that representations of semantically similar discourse segments (e.g., a sad tone in audio and sad imagery in vision) are **close** in the shared embedding space.
- **Separates Dissimilar Representations**:

1. Pushes apart representations of semantically different segments (e.g., a happy tone in audio vs. a sad expression in vision), ensuring the model captures distinct meanings.



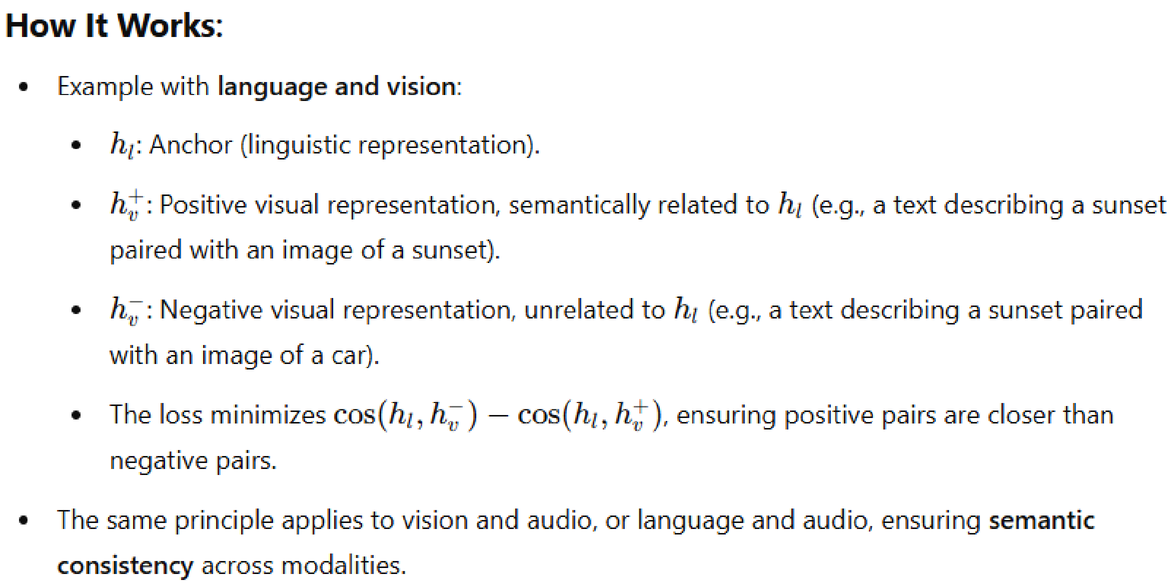
$$L^l_{\text{trip}} = \sum_{m\in\{v,a\}} \max\left(\cos\left(h_l, h_m^-\right) - \cos\left(h_l, h_m^+\right) + \text{margin}, 0\right),$$

$$L^v_{\text{trip}} = \sum_{m\in\{l,a\}} \max\left(\cos\left(h_v, h_m^-\right) - \cos\left(h_v, h_m^+\right) + \text{margin}, 0\right),$$

$$L^a_{\text{trip}} = \sum_{m\in\{l,a\}} \max\left(\cos\left(h_v, h_m^-\right) - \cos\left(h_v, h_m^+\right) + \text{margin}, 0\right).$$

Based on formulas the total cosine triple margin loss is represented as follows:

$$L_{\text{trip}} = L^l_{\text{trip}} + L^v_{\text{trip}} + L^a_{\text{trip}}.$$



How It Works:

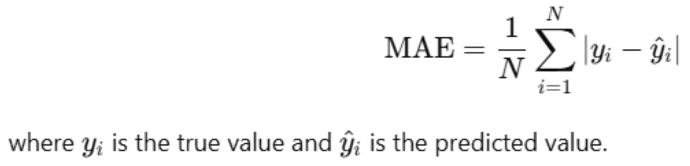
- Example with **language and vision**:
  - h_l: Anchor (linguistic representation).
  - h_v^+: Positive visual representation, semantically related to h_l (e.g., a text describing a sunset paired with an image of a sunset).
  - h_v^-: Negative visual representation, unrelated to h_l (e.g., a text describing a sunset paired with an image of a car).
  - The loss minimizes $\cos(h_l, h_v^-) - \cos(h_l, h_v^+)$, ensuring positive pairs are closer than negative pairs.
- The same principle applies to vision and audio, or language and audio, ensuring **semantic consistency** across modalities.

- Here “ *margin = 1* ” is a boundary parameter.

Evaluation Indices

1. Mean Absolute Error (MAE)

- **Definition**: MAE is a common metric used to evaluate the performance of regression models. It measures the average magnitude of the errors in a set of predictions, without considering their direction. It's calculated as:



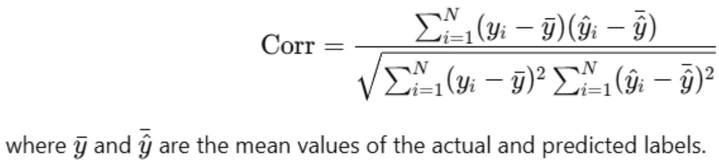
$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

where y_i is the true value and $\hat{y}_i$ is the predicted value.

- **Why it's used**: MAE is used here because the task is regressive in nature (predicting continuous values, such as emotion intensity). MAE provides a clear and interpretable metric of the model's accuracy by directly measuring how far off the predictions are from the true values, on average. It is preferred in scenarios where we want to understand the overall prediction error without penalizing large mistakes more heavily (unlike RMSE, for example).

2. Pearson Correlation Coefficient (Corr)

- **Definition**: The Pearson correlation coefficient is a measure of the linear relationship between two variables. It ranges from -1 to 1, where:
    - 1 indicates a perfect positive linear relationship,
    - -1 indicates a perfect negative linear relationship,
    - 0 indicates no linear relationship.



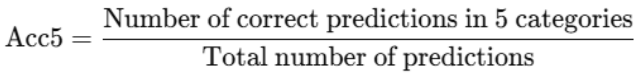
$$\mathrm{Corr} = \frac{\sum_{i=1}^{N}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2 \sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}})^2}}$$

where $\bar{y}$ and $\bar{\hat{y}}$ are the mean values of the actual and predicted labels.

- **Why it's used**: The Pearson correlation is important here because it measures how well the predicted values align with the true values, regardless of the scale. In a regression task, the correlation helps assess the strength of the linear relationship between predicted and actual values. A high Pearson correlation indicates that the model is making predictions that are consistently in the right direction and closely following the true values.

3. Accuracy (Acc5)

- **Definition**: In a classification setting, accuracy measures the proportion of correct predictions. For **five-class classification accuracy (Acc5)**, the task involves predicting one of five emotion classes (e.g., ranging from negative to positive emotions). The accuracy is defined as:



$$\mathrm{Acc5} = \frac{\text{Number of correct predictions in 5 categories}}{\text{Total number of predictions}}$$

The emotion labels might range from -2 (negative emotion) to +2 (positive emotion), and the model’s task is to correctly categorize each sample into one of the five classes.

- **Why it's used**: **Acc5** is used to assess how well the model is performing in predicting the correct emotion class out of the five possible categories. It is important for evaluating the model’s classification performance when it needs to make discrete predictions based on continuous emotional intensity levels.

4. Accuracy (Acc-2)

- **Definition**: **Two-class classification accuracy (Acc-2)** simplifies the problem into two classes—positive (p) and negative (g) emotions. This accuracy is calculated in a similar manner to Acc5 but focuses only on two categories:

$$\text{Acc-2} = \frac{\text{Number of correct predictions (positive/negative)}}{\text{Total number of predictions}}$$

- **Why it's used**: **Acc-2** is often used to evaluate the model when it is simplified into binary classification, such as determining whether the emotion is positive or negative.

5. F1-Score

- **Definition**: The **F1-score** is the harmonic mean of precision and recall. It is often used for imbalanced class problems where one class might dominate the others. It is calculated as:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

  - **Precision** is the proportion of positive predictions that are actually correct.
  - **Recall** is the proportion of actual positives that were correctly identified.

- **Why it's used**: The **F1-score** is particularly useful in scenarios where the class distribution is imbalanced (e.g., some emotions might occur much more frequently than others), and it helps to assess the balance between precision and recall. In emotion detection, we might be more interested in ensuring that we don't miss out on positive emotion cases (high recall) and that when the model predicts positive emotion, it is correct (high precision). The F1-score gives a balanced measure of both.

Read Section 4.3 and 4.4 from paper

Hyperparameters	Best Values
CMD K (value of K in Similarity Loss)	5
Batch_size	16
α (weight of L_{diff})	0.4
β (weight of L_{sim})	0.8
γ (weight of L_{recon})	0.4
η (weight of L_{trip})	0.01
Drop (from dropout layer)	0.1
Hid (Hidden layer size of representation Network)	256
P_h (Hidden layer size of predictive Network)	64
Epochs	20
Learning rate (with Adam)	0.0001

Architecture

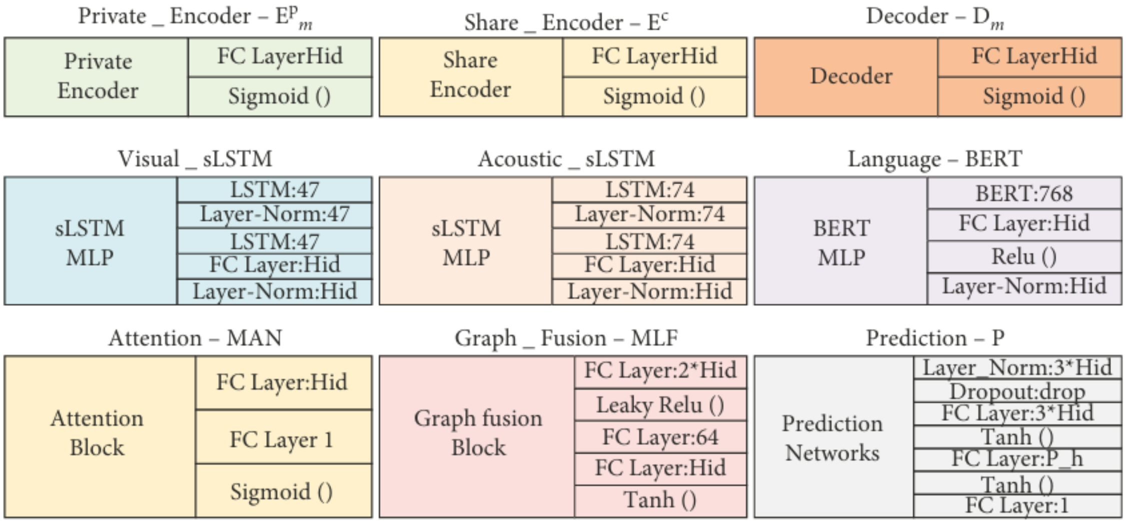


Figure 3: The parameter setting of modules.