CMAD: Correlation-Aware and Modalities-Aware Distillation for Multimodal Sentiment Analysis with Missing Modalities

Supplementary Material

6. Additional Implementation Details

6.1. Teacher Model

In the CMAD framework, both the teacher and student models have identical architectures. After extracting features X_i from each modality, these features are passed through a Conv1D layer to capture contextual and temporal information. This step also normalizes the representations to a uniform length and dimensionality, specifically a length T of 50 and a hidden dimension D. The processed features from each modality are then fed into a Perceiver [12] network to generate modality-specific representations. The Perceiver architecture, illustrated in Fig. 4a, consists of learnable units that learn representations using both cross-modal and self-attention mechanisms [37]. Specifically, the Perceiver layers employed on MOSEI, IEMOCAP, CHERMA, MUStARD and UR-FUNNY datasets are 5, 4, 4, 3 and 6, respectively. Once the modality-specific representations are obtained, they are passed through a two-layer Transformer Encoder followed by a linear layer for feature fusion. The fused representations are then fed into a fully connected layer to perform classification and make final predictions.

To further clarify this process, we outline the steps in the teacher model in detail. Given the multimodal input $x = \{X_1, X_2, ..., X_m\}$, where each modality $X_i \in \mathcal{R}^{B \times T_i \times D_i}$, the features are first processed through a Conv1D layer with a kernel size of 3×3 to capture contextual and temporal relationships. This is represented as:

$$X_i^c = Conv1D(X_i). \tag{16}$$

These transformed representations are then passed through modality encoders, P_i^t, which consist of learnable units, denoted as $E_i^t \in \mathcal{R}^{B \times D}$ for P_i^t, cross-modal and self-attention blocks. Specifically, each Perceiver layer consists of alternating cross-attention and self-attention blocks as in [12]. In the cross-attention block, the query is set as E_i^t, and the key and value are set as X_i^c, formulated as:

$$H_{ca} = \text{Softmax}(\frac{E_i^t W_{Q_e} W_x^\top (X_i^c)^\top}{\sqrt{d_k}}) X_i^c W_{V_x}, \tag{17}$$

where $W_{Q_e}, W_x^\top, W_{V_x} \in \mathcal{R}^{D \times D}$ are trainable weights, $H_{ca} \in \mathcal{R}^{B \times D}$. This is followed by a layer normalization step:

$$\hat{H}_{ca} = LN(E_i^t, H_{ca}) + E_i^t, \tag{18}$$

and a feed-forward layer:

$$H_{ff} = LN(W_{ff} \hat{H}_{ca}) + \hat{H}_{ca}, \tag{19}$$

where $W_{ff} \in \mathcal{R}^{B \times B}$ are trainable weights. The self-attention block follows a similar structure, but the query, key, and value are set to the same input, e.g. H_{ff}. The output of the Perceiver is a modality-specific representation $X_i' \in \mathcal{R}^{B \times D}$.

The final step involves fusing the modality-specific representations. These outputs are concatenated and passed through a two-layer transformer encoder, followed by a linear layer for fusion, as shown in:

$$E^t = W(\text{Transformer Encoder}([X_1', ..., X_m'])) + b, \tag{20}$$

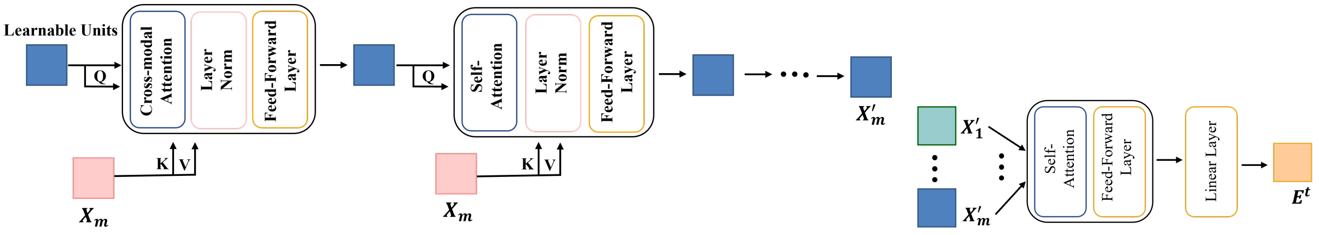
where $[\cdot]$ denotes the concatenation operation, W and b are weights and bias in linear layer.

6.2. Training Configurations

All models were implemented using the PyTorch framework [28] and trained on a GTX 3090 GPU. The teacher and student models for the MOSEI, MUStARD, and UR-FUNNY datasets were trained using the AdamW optimizer [49]. For these datasets, the teacher model had a hidden dimension of $D = 96$ (MOSEI) and $D = 64$ (MUStARD, UR-FUNNY) with a learning rate of $2e - 5$. The student model settings were as follows: MOSEI: $G = 20$, $\sigma = 0.1$, $\gamma = 1$, $\phi = 7.0$, and batch size of 128. MUStARD: $G = 40$, $\sigma = 0.1$, $\gamma = 1$, $\alpha = 0.9$, and batch size of 64. UR-FUNNY: $G = 20$, $\sigma = 0.1$, $\gamma = 1$, $\alpha = 0.8$, and batch size of 96. For IEMOCAP and CHERMA, both teacher and student models were trained using the Adam optimizer [15]. For IEMOCAP, the teacher model had $D = 30$ and a learning rate of $1e - 3$. The student model was trained with a batch size of 120, $\gamma = 0.7$, $\alpha = 0.9$, $G = 10$, and $\sigma = 1$. For CHERMA, the teacher model had a learning rate of $2e - 5$ and $D = 1024$. The student model used a learning rate of $2e - 5$, a batch size of 400, $\gamma = 1$, $\alpha = 0.7$, $G = 5$, and $\sigma = 1$.

6.3. Additional MAR Details

Due to space limitations in the main text, we provide the detailed framework and pseudo-code of the MAR module in Figure 5 and Algorithm 1. All referenced equations can be found in Section 3.4 of the main manuscript, where the MAR module is discussed in detail. We also present a brief summary of the key equations involved in these components in Table 5.



(a) The structure of the modality process P_i^w.

(b) The structure of the fusion network F^w.

Figure 4. The structure of P_i^w and F^w in CMAD.

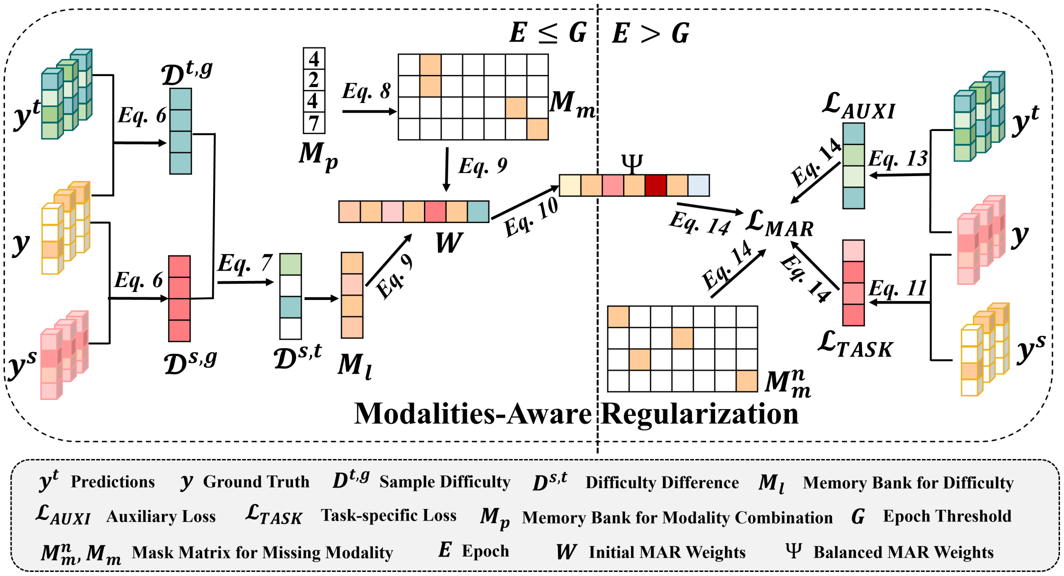


Figure 5. The framework of proposed MAR module.

7. Additional Experiments

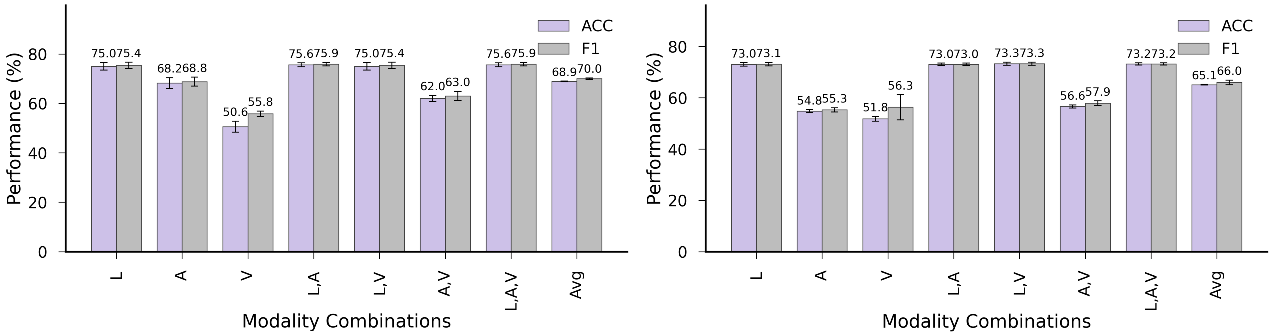
7.1. Additional Results on MUStARD and UR-FUUNY Datasets

We present additional comparisons of the model performances on the MUStARD and UR-FUNNY datasets, as shown in Table 6. Our findings indicate that the models' performance on these two datasets shares something common with the performance on the MOSEI dataset. For instance, across all these datasets, the language modality consistently exhibits the strongest representation ability. When the language modality is missing, the performance of the TEA model significantly drops. On the other hand, MMANet and CMAD$_\mathrm{w}$, which focus on the difficult modality combinations, perform worse in combinations related to the language modality compared to the teacher model. However, CMAD, which consider all modality combinations, maintains a more consistent representation. Additionally, we observe that including sample difficulty (CMAD$_\mathrm{t}$) leads to a noticeable decrease in overall performance. Specifically, in the MUStARD dataset, performance dropped by 4.85% and 4.6% on two key metrics. Finally, while the customized models, such as MPLMM, showed promising results in certain modality combinations (e.g. '{V}' on both datasets), they require multiple runs and do not account for the varying importance of different modality combinations. As a result, their overall performance still lags behind that of our CMAD model.

To further assess the robustness of CMAD to training randomness, we conducted five independent runs on both MUStARD and UR-FUNNY using different random seeds. As illustrated in Figure 6, the results are consistent across runs, with low variance indicated by narrow error bars, demonstrating CMAD's stability under different initialization conditions.

7.2. Additional Results of Intra-modality Missingness

To further evaluate the general applicability of CMAD in more realistic scenarios, we follow prior works such as CorrKD [17, 18] and assess model performance under intra-modality missingness. Specifically, we simulate this condi-



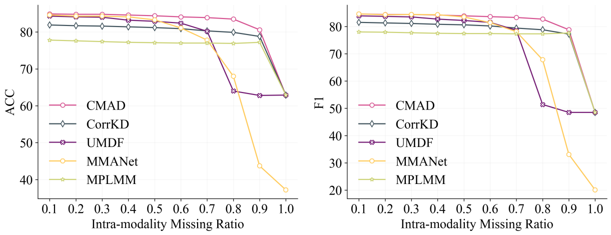
(a) Five runs on MUStARD.

(b) Five runs on UR-FUNNY.

Figure 6. Visualization of performance on MUStARD and UR-FUNNY.

```
Algorithm 1: Modalities-Aware Regularization
Input: Student model prediction $y^s$, teacher model
       prediction $y^t$, ground-truth label $y$, modality
       combination used in the student input $\Delta_s$,
       epoch threshold $G$
Output: Modalities-Aware Regularization (MAR)
        loss $\mathcal{L}_{MAR}$
Initialize memory banks $M_l$ (difficulty memory) and
  $M_p$ (modality pattern memory);
if Epoch $E \leq G$ then
    Estimate sample difficulty $\mathcal{D}^{s,t}$ using $y^s$, $y^t$, and
      $y$ via Eq. 6 and Eq. 7;
    Store $\Delta_s$ into $M_p$ and $\mathcal{D}^{s,t}$ into $M_l$;
    Set all weights in $\Psi$ to 1.0;
else
    Compute weights $W$ based on $M_p$ and $M_l$ using
      Eq. 8 and Eq. 9;
    Normalize and refine $W$ to obtain the adjusted
      weights $\Psi$ via Eq. 10;
Compute task loss $\mathcal{L}_{TASK}$ and auxiliary loss
  $\mathcal{L}_{AUXI}$ using $y^s$, $y^t$, and $y$ according to Eq. 11–13;
Aggregate the final loss $\mathcal{L}_{MAR}$ by weighting
  $\mathcal{L}_{TASK}$ and $\mathcal{L}_{AUXI}$ with $\Psi$ using Eq. 14;
return $\mathcal{L}_{MAR}$;
```

tion by dropping a portion of the frame-level features within each modality sequence on the MOSEI dataset. The drop ratio $p \in \{0.1, 0.2, ..., 1.0\}$ controls the severity of missing data, where $p = 1.0$ indicates complete feature removal in all modalities, and $p = 0.1$ denotes a 10% loss of representation length in each modality. For a fair comparison, we reproduce CorrKD [18], UMDF [17], MMANet [43], and MPLMM [8] under identical settings. As illustrated in Figure 7, CMAD consistently outperforms all baselines in terms of both accuracy and F1 score across varying levels of intra-modality missingness, highlighting its strong robustness and adaptability in challenging conditions.



(a) 'ACC' performance.

(b) 'F1' performance.

Figure 7. Comparison results of intra-modality missingness on MOSEI.

7.3. Analysis on the Consistency in CAFD

Here we provide additional discussion on the consistency within the CAFD module. Specifically, we analyze four scenarios: (1) only use the feature consistency calculated through $\mathcal{L}^{MSE}$ in Eq. 1; (2) only use the correlation consistency calculated with $\mathcal{L}^{S_p}$ in Eq. 3 and $\mathcal{L}^{S_a}$ in Eq. 4 through $\mathcal{L}^{S_p} + \mathcal{L}^{S_a}$; (3) use the $\mathcal{L}_{CAFD}$ in Eq. 5; and (4) direct transfer the teacher-teacher correlations $R^{t,t}$ to student-student correlations $R^{s,s}$, where $R^{s,s} \in \mathcal{R}^{B \times B}$ is calculated between the student representations E^s using Eq. 2.

The results are shown in Tables 7, 8, 9 and 10. Here $\mathcal{L}_{mse}$ and $\mathcal{L}_{sim}$ represent the feature consistency loss ($\mathcal{L}^{MSE}$), and correlation consistency losses ($\mathcal{L}^{S_p}$ and $\mathcal{L}^{S_a}$), respectively. A '✓' indicates the application of these losses, while a '×' indicates a direct transfer of teacher sample correlations to the student model. 'MSE' and 'SIMS' represent the corresponding metric values. 'MSE' is the average of all $\mathcal{L}^{MSE}$ values, while 'SIMS' is the averaged sum of $\mathcal{L}^{S_p}$ and $\mathcal{L}^{S_a}$. All results are the average values across all seven possible combinations of missing modalities. Lower values for

Indices	Equations	Description
Eq. 6	$\mathcal{D}_i^{w,g} = \mathcal{L}_{TASK}(y_i^w, y_i)$	Get the sample difficulty from the model w.
Eq. 7	$\mathcal{D}_i^{s,t} = \begin{cases} 0, & if \mathcal{D}_i^{s,g} < \mathcal{D}_i^{t,g} \\ \mathcal{D}_i^{s,g} - \mathcal{D}_i^{t,g}, & otherwise \end{cases}$	Get the sample difficulty difference.
Eq. 8	$M_m(i,j) = \begin{cases} 1, & if M_p(i) = j \\ 0, & otherwise \end{cases}$	Get the missing modality mask matrix.
Eq. 9	$W(j) = \begin{cases} 0, & if N_m(j) = 0 \\ \frac{(M_l \times M_m)(j)}{N_m(j)}, & otherwise \end{cases}$	Get the initial MAR weights.
Eq. 10	$\Psi(j) = \rho(\frac{W(j)}{max(W)})^2$	Get the balanced MAR weights.
Eq. 11	$\mathcal{L}_{TASK}(y^w, y) = \begin{cases} MAELoss(y^w, y), & if k = 1 \\ CrossEntropy(y^w, y), & if k > 1 \end{cases}$	Get the task-specific loss.
Eq. 12	$DKD(y^s, y^t) = \alpha KL(b^t \Vert b^s) + (1-\alpha) KL(\hat{p}^t \Vert \hat{p}^s)$	Get the auxiliary loss for classification task.
Eq. 13	$\mathcal{L}_{AUXI}(y^s, y^t) = \begin{cases} MSE(\frac{y^s}{\phi}, \frac{y^t}{\phi}) * (\phi)^2, & if k = 1 \\ DKD(y^s, y^t), & if k > 1 \end{cases}$	Get the auxiliary loss for all tasks.
Eq. 14	$\mathcal{L}_{MAR} = \mathcal{L}_{TASK} * M_m^n * \Psi + \gamma \mathcal{L}_{AUXI} * M_m^n * \Psi$	Get the final MAR loss.

Table 5. Equations in Modalities-Aware Regularization module.

Set	Models	**Modalities**							Avg.
		L	A	V	L,A	L,V	A,V	L,A,V	
		Results on MUStARD							
C	MPLMM[8]	67.6/68.1	61.8/61.8	**61.8/61.8**	67.6/68.1	67.6/68.1	**61.8**/61.8	67.6/68.1	65.1/65.4
	UMDF[17]	72.1/72.8	66.2/67.5	51.5/60.6	72.1/72.8	72.1/72.8	58.8/59.1	72.1/72.8	66.4/68.3
	CorrKD[18]	72.1/72.1	66.2/67.9	45.6/50.2	72.1/72.1	72.1/72.1	57.4/59.0	72.1/72.1	65.3/66.5
U	TEA	70.6/71.0	52.9/57.4	52.9/57.4	70.6/71.0	70.6/71.0	52.9/57.4	70.6/71.0	63.0/65.2
	MMANet[43]	69.1/69.1	64.7/66.3	47.1/53.2	69.1/69.1	69.1/69.1	60.3/61.4	69.1/69.1	64.1/65.3
	CMAD$_w$	70.6/70.7	64.7/66.3	45.6/49.2	70.6/70.7	69.1/69.4	60.3/**62.3**	70.6/70.7	64.5/65.6
	CMAD$_t$	72.1/72.1	57.4/59.0	47.1/54.6	72.1/72.1	72.1/72.1	55.9/57.3	72.1/72.1	64.1/65.6
	CMAD	**76.5/76.7**	**67.6/68.1**	47.1/54.6	**76.5/76.7**	**76.5/76.7**	**61.8/62.3**	**76.5/76.7**	**68.9/70.2**
		Results on UR-FUNNY							
C	MPLMM[8]	71.7/71.8	52.3/54.3	**53.0/54.9**	72.4/72.5	71.7/71.8	54.8/55.2	72.4/72.5	64.0/64.7
	UMDF[17]	71.8/71.8	53.7/54.2	52.4/**54.9**	72.1/72.2	71.8/71.9	56.1/56.2	72.2/72.3	64.3/64.8
	CorrKD[18]	72.4/72.4	53.8/54.0	52.0/54.5	72.5/72.5	72.4/72.4	**56.2**/56.3	72.4/72.4	64.5/64.9
U	TEA	72.7/72.7	52.7/54.2	52.7/54.2	72.7/72.7	72.7/72.7	52.7/54.2	72.7/72.7	64.1/64.8
	MMANet[43]	71.0/71.2	53.1/53.2	52.1/52.8	70.7/70.9	70.9/71.1	55.3/55.7	70.6/70.8	63.4/63.7
	CMAD$_w$	71.4/71.5	53.6/53.6	52.5/53.5	71.8/71.9	71.4/71.5	55.0/55.3	71.5/71.6	63.9/64.1
	CMAD$_t$	72.5/72.5	53.0/53.4	51.7/53.0	72.5/72.5	72.3/72.3	55.7/55.8	72.3/72.3	64.3/64.6
	CMAD	**72.8/72.9**	**54.2/54.4**	52.5/53.4	**73.1/73.2**	**73.0/73.0**	**56.2/56.4**	**73.3/73.3**	**65.0/65.2**

Table 6. Performance comparison under different modality combinations on MUStARD and UR-FUNNY datasets. 'ACC/F1' is reported.

both metrics are preferred, as they represent the differences in features and correlations across samples.

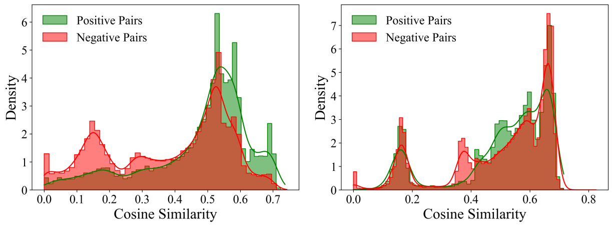
From Tables 7 and 8, we note a notable trend: modality combinations with lower metric values, indicating better consistency, consistently perform better in predictive tasks. This trend underscores that robust consistency across modality combinations positively impacts model accuracy. Additionally, we identify a consistent pattern: Using only $\mathcal{L}_{mse}$ yields the best feature consistency, while $\mathcal{L}_{sim}$ alone achieves near-optimal correlation consistency. However, each performs poorly when evaluated on the other metric, highlighting the limitations of relying on a single consistency. A balanced approach, combining both losses, yields improved performance across metrics. Moreover, direct correlation transfer performs the worst on both feature and correlation consistency. These findings highlight the neces-

Losses				Modalities							Avg.
L_{mse}	L_{sim}	MSE	SIMS	L	A	V	L,A	L,V	A,V	L,A,V	
✓	✓	✓		**0.29**	0.33	0.36	**0.28**	**0.30**	0.36	**0.29**	0.32
✓		✓		**0.29**	**0.32**	**0.34**	0.29	**0.30**	**0.35**	0.31	**0.31**
	✓	✓		0.36	0.45	0.45	0.36	0.36	0.46	0.37	0.40
	×	✓		0.67	0.76	0.70	0.67	0.64	0.72	0.63	0.68
✓	✓		✓	**0.63**	2.34	2.39	**0.63**	0.71	2.29	0.76	1.39
✓			✓	1.00	2.45	2.29	1.12	1.18	2.30	1.29	1.66
	✓		✓	0.66	**2.18**	**2.21**	0.68	**0.68**	**2.15**	**0.68**	**1.32**
	×		✓	2.92	2.52	2.68	2.87	3.05	2.55	2.96	2.79

Table 7. Consistency results of different consistency losses used in $\mathcal{L}_{CAFD}$ on MOSEI dataset. Lower value is better.

Losses				Modalities							Avg.
L_{mse}	L_{sim}	MSE	SIMS	L	A	V	L,A	L,V	A,V	L,A,V	
✓	✓	✓		1.36	1.23	**1.44**	1.23	1.31	1.20	1.21	1.28
✓		✓		**1.17**	**1.12**	1.49	**0.94**	**1.09**	**1.07**	**0.90**	**1.11**
	✓	✓		2.73	2.36	2.41	2.75	2.70	2.41	2.73	2.58
	×	✓		3.08	2.62	2.40	3.30	3.27	2.86	3.47	3.00
✓	✓		✓	1.70	**1.65**	**2.34**	1.18	**1.45**	**1.44**	1.02	**1.54**
✓			✓	1.78	1.80	2.75	**1.15**	1.54	1.58	**1.00**	1.66
	✓		✓	**1.64**	1.69	**2.34**	1.20	1.46	1.52	1.09	1.56
	×		✓	3.17	3.20	2.84	3.29	3.19	3.28	3.36	3.19

Table 8. Consistency results of different consistency losses used in $\mathcal{L}_{CAFD}$ on IEMOCAP dataset. Lower value is better.



(a) Similarity distribution with CL. (b) Similarity distribution w/o CL.

Figure 8. Similarity distribution visualization on MOSEI.

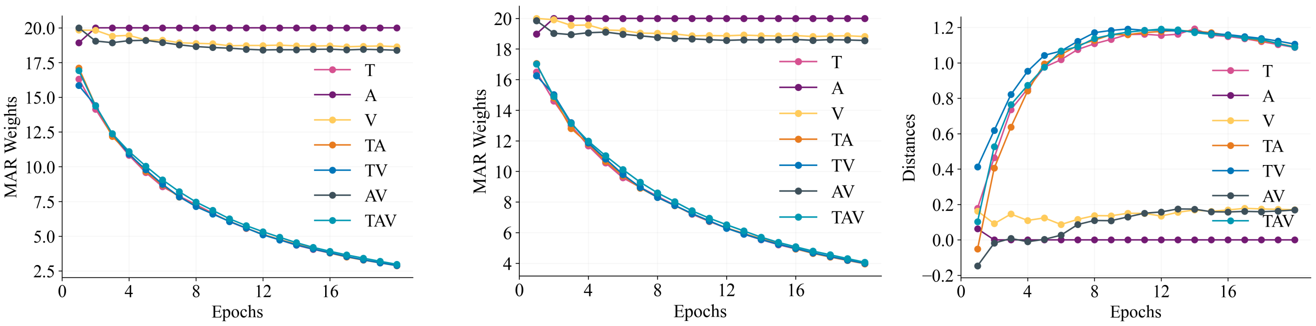
sity of simultaneously considering both feature and correlation consistencies. Additionally, as demonstrated in Tables 9 and 10, focusing exclusively on either feature or correlation consistency results in reduced overall performance.

To further highlight the effectiveness of contrastive learning (CL) in capturing high-level semantics within CAFD, we analyze the similarity distribution between positive and negative pairs on the MOSEI dataset. As shown in Figure 8, the inclusion of CL leads to higher similarity among positive pairs and lower similarity among negative pairs. These results clearly illustrate the positive impact of CL in enhancing semantic alignment and discrimination.

7.4. Analysis on the Loss Components in MAR

In this section, we conduct additional ablation experiments to analyze the significance of each loss component in the CMAD model. As shown in Table 11 and Table 12, removing any of the loss modules results in a decline in overall performance. Specifically, L_{TM}, L_{AM}, L_T, L_A, L_C represents the use of the MAR weights in $\mathcal{L}_{TASK}$, the MAR weights in $\mathcal{L}_{AUXI}$, the $\mathcal{L}_{TASK}$, the $\mathcal{L}_{AUXI}$ and $\mathcal{L}_{CAFD}$, respectively.

In the multimodal regression task on the MOSEI dataset, removing the $\mathcal{L}_{AUXI}$ led to the most significant decrease in F1-score, while removing all MAR weights caused the largest drop in accuracy. This suggests that in the regression task, the $\mathcal{L}_{AUXI}$ is crucial for transferring knowledge from the teacher model to help the student model learn consistent representations, while the MAR weights are essential for guiding the model in understanding the importance of different modality combinations, thereby enhancing its performance. For multimodal classification task on the IEMOCAP dataset, removing the $\mathcal{L}_{AUXI}$ resulted in the largest decrease in accuracy, while the absence of the $\mathcal{L}_{TASK}$ led to the most substantial decline in the F1-score. This indicates that in classification tasks, it is vital to consider the probability distribution over the categories. Relying solely on guidance from the teacher model or focusing only on the $\mathcal{L}_{TASK}$ leads to a loss of critical information, thereby im-



(a) Visualization of MAR weights with $\mathcal{D}^{t,g}$.
(b) Visualization of MAR weights without $\mathcal{D}^{t,g}$.
(c) Visualization of MAR weights difference.

Figure 9. Visualization of MAR weights and MAR weights distances on MOSEI dataset.

Losses		**Modalities**							Avg.
L_{mse}	L_{sim}	L	A	V	L,A	L,V	A,V	L,A,V	
✓	✓	**86.1/86.0**	63.0/**60.8**	**65.7/64.4**	**86.3/86.2**	**86.4/86.4**	**65.6/64.8**	**86.1/86.1**	**77.03/76.39**
✓		85.7/85.6	63.5/59.2	63.8/63.7	85.6/85.5	85.8/85.8	64.9/64.4	85.7/85.8	76.43/75.71
	✓	85.2/85.1	63.5/59.6	64.4/64.1	85.1/84.9	86.1/86.0	64.8/64.2	85.6/85.6	76.39/75.64
	×	85.6/85.5	**63.6**/59.0	63.3/63.3	85.4/85.2	85.5/85.4	64.7/64.2	85.5/85.5	76.23/75.43

Table 9. Performance of different consistency losses used in $\mathcal{L}_{CAFD}$ on MOSEI dataset. Higher value is better.

Losses		**Modalities**							Avg.
L_{mse}	L_{sim}	L	A	V	L,A	L,V	A,V	L,A,V	
✓	✓	**79.5/78.7**	**79.2/76.7**	73.6/70.6	**81.9/81.3**	**79.7/79.1**	79.0/77.3	**81.7/81.2**	**79.23/77.84**
✓		78.2/77.0	78.5/75.9	73.5/70.1	81.2/80.1	77.7/76.6	78.6/76.8	80.8/79.8	78.36/76.61
	✓	78.6/77.4	78.4/75.9	**74.3/71.2**	81.1/80.2	78.7/77.8	**79.4/77.8**	81.4/80.6	78.84/77.27
	×	77.7/77.1	77.5/74.6	73.3/70.5	79.9/79.3	77.9/77.3	77.7/76.0	80.4/79.9	77.78/76.39

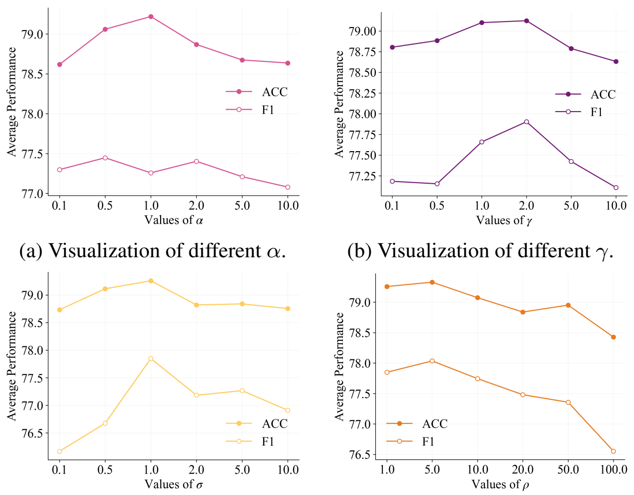
Table 10. Performance of different consistency losses used in $\mathcal{L}_{CAFD}$ on IEMOCAP dataset. Higher value is better.

Losses					**Modalities**						
L_{TM}	L_{AM}	L_T	L_A	L_C	L	A	V	L,A	L,V	A,V	L,A,V
✓	✓	✓	✓	✓	**86.1/86.0**	63.0/**60.8**	**65.7/64.4**	**86.3/86.2**	**86.4/86.4**	65.6/**64.8**	**86.1/86.1**
✓	✓	✓	✓		85.6/85.4	63.8/60.4	63.3/62.6	85.5/85.4	85.4/85.3	64.2/63.6	85.7/85.6
		✓	✓	✓	85.3/85.3	58.9/59.5	63.8/63.0	85.1/85.1	85.6/85.5	64.1/64.2	85.6/85.6
✓		✓	✓	✓	85.6/85.4	63.8/57.8	64.6/60.3	85.6/85.5	85.8/85.7	66.6/64.3	86.0/86.0
	✓	✓	✓	✓	85.5/85.4	64.0/58.7	64.3/61.8	85.6/85.6	86.1/86.0	65.7/64.3	85.9/86.0
	✓		✓	✓	85.7/85.6	60.0/60.0	64.3/64.1	85.6/85.4	85.6/85.6	65.3/64.7	85.9/85.9
✓		✓		✓	85.0/84.7	**64.4**/54.5	64.1/60.3	85.2/85.1	85.5/85.3	**66.7**/64.1	85.8/85.7

Table 11. Ablation of different loss components used in $\mathcal{L}_{total}$ on MOSEI dataset.

Losses					**Modalities**						
L_{TM}	L_{AM}	L_T	L_A	L_C	L	A	V	L,A	L,V	A,V	L,A,V
✓	✓	✓	✓	✓	**79.5/78.7**	**79.2/76.7**	73.9/**70.6**	**81.9/81.3**	**79.7/79.1**	**79.0/77.3**	**81.7/81.2**
✓	✓	✓	✓		78.9/77.8	**79.2**/75.9	72.8/69.2	81.4/80.4	79.0/77.9	78.6/76.2	81.5/80.4
		✓	✓	✓	78.6/77.2	77.9/73.0	75.3/70.1	80.6/79.5	**79.7**/78.6	78.9/75.9	81.5/80.6
✓		✓	✓	✓	78.7/76.7	76.7/72.2	73.5/69.5	80.6/78.7	79.2/77.5	78.5/75.9	81.3/79.9
	✓	✓	✓	✓	78.8/77.6	77.7/74.8	74.0/69.3	80.5/79.3	78.9/77.7	77.7/75.3	81.1/80.0
	✓		✓	✓	78.4/75.6	78.1/73.7	**75.5**/68.5	80.4/78.1	78.8/76.3	78.8/74.9	80.8/78.6
✓		✓		✓	78.1/77.3	78.0/75.6	73.7/70.5	79.5/78.8	77.9/77.3	78.4/76.8	80.0/79.4

Table 12. Ablation of different loss components used in $\mathcal{L}_{total}$ on IEMOCAP dataset.



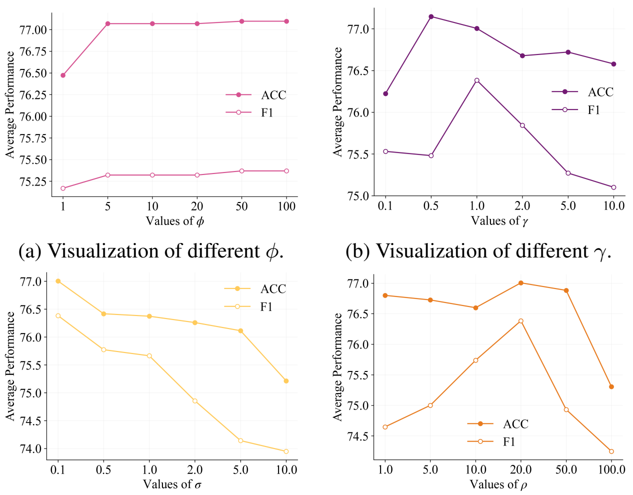
(a) Visualization of different α.

(b) Visualization of different γ.

(c) Visualization of different σ.

(d) Visualization of different ρ.

Figure 10. Visualization of the performance of different hyper-parameters on IEMOCAP dataset.



(a) Visualization of different ϕ.

(b) Visualization of different γ.

(c) Visualization of different σ.

(d) Visualization of different ρ.

Figure 11. Visualization of the performance of different hyper-parameters on MOSEI dataset.

pairing the model's performance.

7.5. Analysis on Sample Difficulty

To clearly illustrate the effects of sample difficulty, we analyze the MAR weights from the MOSEI dataset before the starting epoch 20. This includes weights that eliminates sample difficulty, those that do not, and the differences between them, as shown in Fig. 9. Our findings indicate that, regardless of whether sample difficulty is eliminated, the weights for each modal combination change similarly over the epochs. However, the range of decline varies between the two conditions. Without eliminating sample difficulty, the weights for hard combinations decrease at a slower rate, ultimately stabilizing around a value of 4, compared to around 2.5 when sample difficulty is eliminated. Moreover, for easy modality combinations, the difference in weights initially increases, then decreases. In contrast, weights for difficult modality combinations gradually increase, leading to a shift in the relative magnitudes of MAR weights, from an 8-fold difference to a 4-fold difference. This pattern indicates that, when sample difficulty is eliminated, the model learns a more consistent relationship between modality combinations, resulting in a more consistent representation that better adapts across all modality configurations.

7.6. Analysis on Hyper-parameters

To better understand the impact of different hyper-parameters, we conduct additional experiments across MOSEI and IEMOCAP datasets. The results from all seven possible combinations of missing modalities are averaged for comparison, as shown in Fig. 10 and Fig. 11. Our findings are summarized as follows: (1) Increasing α, which amplifies the loss of the target class for classification tasks, does not always lead to better performance. In fact, the probabilities from non-target class samples also play a crucial role. (2) Increasing ϕ in the auxiliary loss $\mathcal{L}_{AUXI}$ for regression task can improve model performance. (3) Moderately increasing the weight of $\mathcal{L}_{AUXI}$, represented by γ, enhances performance. However, if γ exceeds the weight of the main task loss $\mathcal{L}_{TASK}$, it can lead to performance degradation. (4) Similarly, excessively increasing the weight of $\mathcal{L}_{CAFD}$, denoted by σ, can negatively affect results. (5) Increasing the value of MAR weights Ψ too much, represented by ρ, can also degrade performance.