MEME: Lightweight Hierarchical Mixture-of-Experts for Unified Affective Computing

Yinan Zhang, Haoyu Zhang, Tianshu Yu*
School of Data Science
The Chinese University of Hong Kong, Shenzhen
{yinanzhang1,haoyuzhang3}@link.cuhk.edu.cn
yutianshu@cuhk.edu.cn

Abstract

Multimodal sentiment analysis (MSA) and emotion recognition (ER) are closely related affective understanding tasks, yet they are usually studied separately due to differences in label space, task formulation, dataset distribution, and modality dependence. In this paper, we propose MEME (**M**ixture-of-**E**xperts for **M**ultimodal sentiment analysis and **E**motion recognition), a lightweight hierarchical MoE framework for unified multimodal affective computing across MSA, conversational emotion recognition, and dynamic facial expression recognition. MEME operates on frozen text, visual, and audio features, compresses variable-length modality sequences with learned-query attention pooling, and processes them with a shared hierarchical MoE backbone. Each block first applies hard modality experts for modality-specific refinement and then task-conditioned cross experts for task-aware multimodal interaction. A `[TASK]` token provides an explicit task anchor for routing, while modality dropping regularizes unified training. We jointly train MEME on nine affective benchmarks and evaluate all datasets using a single composite-best checkpoint without dataset-wise adaptation. Extensive experiments show that MEME outperforms strong task-specific baselines and recent large-model-based unified methods on most benchmarks, while maintaining favorable efficiency, robustness and generalization.

1 Introduction

Understanding human affect from multimodal cues remains a central goal of affective computing. In real-world scenarios, sentiment and emotion are often expressed jointly through language, vocal prosody, and facial or visual behavior, which has motivated extensive work on multimodal sentiment analysis (MSA) and emotion recognition (ER) [1, 2]. MSA is typically formulated as regression over continuous sentiment intensity [3, 4], whereas ER is usually formulated as classification over discrete affective categories [5, 6]. ER itself spans conversational settings (ERC), which rely more on dialogue context and speaker interaction [7], and facial settings (FER), which rely more on facial motion and nonverbal visual cues [8]. Despite these differences, MSA and ER share part of the underlying affective information and multimodal evidence. For example, CMU-MOSEI [9] provides both sentiment and emotion annotations for the same utterances, and our correlation analysis in Appendix B shows non-trivial label correlations rather than independent supervision signals. Similar observations have also been discussed in previous work [10]. This makes joint modeling of MSA and ER a natural direction rather than an arbitrary combination of unrelated tasks.

Making such unification work well, however, is challenging. MSA and ER differ in task form, label space, dataset distribution, and modality contribution patterns; ERC and FER further impose different

*Correspondence author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

requirements within ER itself. A unified model therefore needs to balance shared representation learning with task-specific specialization under heterogeneous supervision. As a result, most prior work still studies MSA, ERC, and FER separately with task-specific architectures and optimization strategies [3, 4, 6, 11]. UniMSE [10] first explicitly unified MSA and ER at the feature, label, and model levels, and later work explored unified generative or large-model-based affective learning [12–14]. These studies establish the feasibility of unified sentiment–emotion modeling, but they remain largely centered on MSA and ERC, often rely on large-model adaptation or larger computational budgets, and typically do not evaluate a single shared checkpoint across heterogeneous MSA, ERC, and FER datasets.

To address these limitations, we propose MEME, a lightweight hierarchical mixture-of-experts architecture for unified multimodal sentiment analysis and emotion recognition. MEME is jointly trained over nine datasets spanning MSA, ERC, and FER, including MOSI [15], MOSEI [9], SIMS [16], SIMS-V2 [17], MELD [18], IEMOCAP [19], MAFW [20], DFEW [21], and AFEW [22], and is evaluated with one composite-best checkpoint selected by a unified validation criterion. The model operates on frozen text, vision, and audio features, compresses variable-length modality sequences with learned-query attention pooling, and processes them with a shared hierarchical MoE backbone. Each block first applies hard modality experts for modality-specific refinement and then task-conditioned cross experts for task-aware multimodal interaction, with a `[TASK]` token serving as an explicit routing anchor. We further introduce modality dropping as a simple regularizer for unified training, which improves optimization stability and reduces over-reliance on dominant modalities. Although related to modality dropout and missing-modality learning [23–25], our use of modality dropping mainly supports heterogeneous unified learning rather than dedicated modality recovery, while also yielding strong robustness under missing-modality evaluation.

Overall, our work extends unified affective learning from earlier MSA–ERC settings to a broader single-checkpoint framework spanning MSA, ERC, and FER. It provides a compact hierarchical MoE design for heterogeneous sharing across modalities and tasks, and offers a unified alternative to recent large-model-centered approaches to multimodal affective learning.

2 Related Work

2.1 Unified Multimodal Sentiment Analysis and Emotion Recognition

Recent progress in multimodal affective computing has largely been driven by task-specific models. DecAlign [26] improves multimodal sentiment analysis with stronger cross-modal alignment and fusion, LSDGNN [6] enhances contextual reasoning for conversational emotion recognition with graph-based modeling, and CLVSR [27] strengthens visual-temporal representation learning for facial expression recognition. Recent MSA work also studies robustness to incomplete multimodal inputs [28]. Beyond task-specific modeling, several studies have started to explore unified affective learning. UniMSE [10] is an early attempt to jointly model multimodal sentiment analysis and emotion recognition within a shared framework. More recent work increasingly moves toward large-model-based formulations: AffectGPT [29] studies emotion understanding with multimodal large language models, EGMF [14] introduces expert-guided multimodal fusion on top of large language models, and MSE-Adapter [13] equips LLMs with lightweight multimodal adaptation modules for sentiment and emotion tasks. Despite this progress, existing unified formulations still mostly remain within limited affective scopes or depend on large-model adaptation. In contrast, our work targets a broader and more stringent unified setting that jointly covers multimodal sentiment analysis, conversational emotion recognition, and facial/video emotion recognition. It further emphasizes a truly unified paradigm, where a single shared checkpoint is trained across heterogeneous datasets and used for evaluation across both MSA and ER without dataset-wise adaptation.

2.2 Mixture-of-Experts for Multimodal Affective Learning

Mixture-of-experts (MoE) has been widely used in multimodal learning to improve specialization and conditional computation by routing different inputs to different experts [30]. In general multimodal learning, MMoE [31] enhances multimodal models with mixtures of multimodal experts, AVMoE [32] studies expert routing for audio-visual representation learning, and I^2MoE [33] models multimodal interactions with interpretable expert decomposition. In affective computing, EMOE [34] introduces modality-specific dynamic experts for emotion recognition, CAG-MoE [35] uses cross-attention

gated experts for multimodal emotion recognition, and MiSTER-E [36] applies a mixture-of-experts design to conversational emotion recognition. Prior MoE-based multimodal models mainly target individual tasks or relatively homogeneous settings, rather than a unified affective formulation spanning heterogeneous datasets and prediction targets. They also do not explicitly exploit the correlation between sentiment and emotion to encourage expert specialization across both modalities and tasks. MEME instead adopts a hierarchical MoE design for unified affective learning, performing modality-specific refinement before task-conditioned cross-modal interaction so that different experts can specialize in modality-dependent and task-relevant patterns within a unified backbone.

2.3 Modality Dropping and Learning with Missing Modalities

Modality dropping has been used in multimodal learning as a simple way to regularize training and reduce over-reliance on any single modality. Existing work has explored modality dropout or related strategies in affective computing and multimodal representation learning [23, 37]. A recent survey further reviews the broader landscape of multimodal learning with missing modalities, covering problem settings, methodological families, and evaluation protocols [25]. A related but distinct line of work studies dedicated incomplete-modality inference. MMIN [38] addresses emotion recognition with incomplete multimodal data, IMDer [39] studies diffused multimodal emotion recognition under incomplete inputs, and RECAP [40] addresses modality missing in multimodal sentiment analysis. These methods are designed primarily for robustness or recovery when one or more modalities are unavailable at inference time. In our setting, we do not formulate the problem as dedicated missing-modality recovery. Instead, modality dropping is introduced first as a training regularization mechanism for unified heterogeneous learning, helping stabilize joint optimization across datasets and reducing dependence on dominant modalities. The resulting model nevertheless exhibits strong natural robustness under missing-modality evaluation, which is a consequence of the unified training design rather than the primary objective of the method.

3 Method

3.1 Overview

Figure 1 illustrates the overall pipeline of MEME. Given text, visual, and audio inputs, MEME extracts frozen modality features, projects them into a shared hidden space, and compresses each variable-length sequence into a fixed-size token block with learned-query attention pooling [41]. During training, modality dropping randomly masks pooled modality blocks while retaining at least one modality. The retained tokens are concatenated with a global `[CLS]` token and a task token `[TASK]`, augmented with position, modality, and dataset embeddings, and processed by L Hierarchical MoE blocks. Each block performs staged self-attention, hard modality-expert refinement, and task-conditioned cross-expert interaction. The final `[CLS]` state is fed into a dataset-specific prediction head. All datasets share the same backbone, and evaluation uses one composite-best checkpoint selected by a unified validation criterion.

3.2 Multimodal Input and Embedding

Let the pre-extracted feature sequence of modality $m \in \{T, V, A\}$ be

$$X_m \in \mathbb{R}^{T_m \times d_m}, \tag{1}$$

where T, V, and A denote text, vision, and audio, respectively; T_m is the sequence length of modality m; and d_m is the corresponding frozen-encoder feature dimension. In our implementation, the text, visual, and audio encoders are DeBERTa-v3-large [42], SigLIP-SO400M [43], and Whisper-large-v3 [44], respectively.

Each modality is projected into a shared hidden space of width d:

$$S_m = E_m(X_m) \in \mathbb{R}^{T_m \times d}, \tag{2}$$

where E_m is the modality-specific projection layer and S_m is the projected sequence. We then compress S_m into a fixed number of latent tokens:

$$H_m = \mathrm{Pool}_m(S_m; Q_m) \in \mathbb{R}^{n_m \times d}, \tag{3}$$

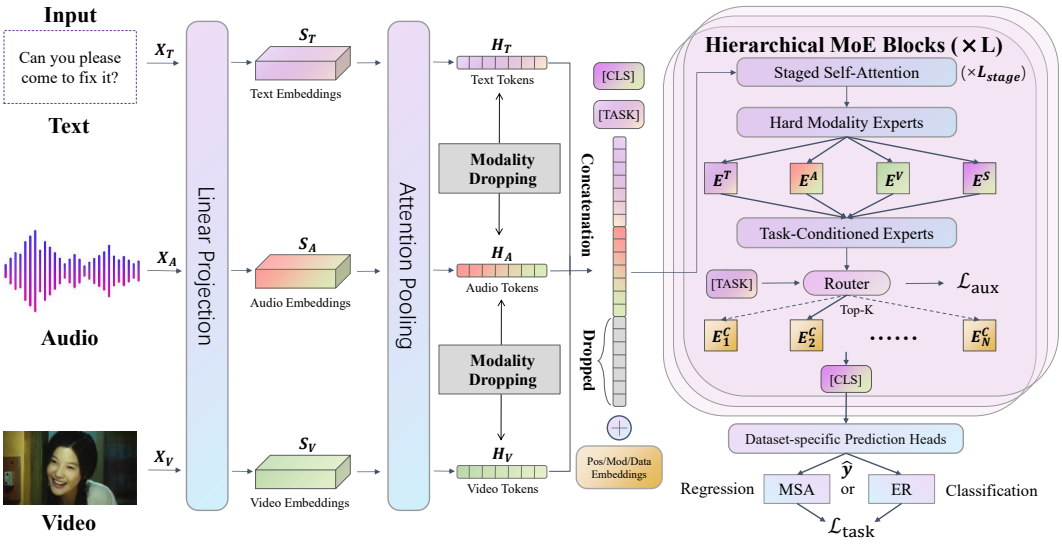


Figure 1: Overview of MEME.

where $Q_m \in \mathbb{R}^{n_m \times d}$ is a set of learnable query vectors, n_m is the token budget for modality m, and $\mathrm{Pool}_m(\cdot)$ is a cross-attention pooling module implemented with multi-head attention followed by LayerNorm [45]. In the main configuration, we use $n_T = 16$ and $n_V = n_A = 8$.

During training, each pooled modality block is independently zeroed with probability p, subject to the constraint that at least one modality is retained. We denote the resulting blocks by $\tilde{H}_T$, $\tilde{H}_V$, and $\tilde{H}_A$. Modality dropping is disabled at inference time. The initial token sequence is constructed as

$$X^{(0)} = \left[h_{\mathrm{CLS}}; h^{\tau}_{\mathrm{TASK}}; \tilde{H}_T; \tilde{H}_V; \tilde{H}_A\right] + E^{\mathrm{pos}} + E^{\mathrm{mod}} + E^{\mathrm{data}}_k \in \mathbb{R}^{S \times d}, \tag{4}$$

where $h_{\mathrm{CLS}} \in \mathbb{R}^d$ is a learnable global prediction token, $h^{\tau}_{\mathrm{TASK}} \in \mathbb{R}^d$ is the learnable task token for task type $\tau \in \{\mathrm{MSA}, \mathrm{ER}\}$, and $S = 2+n_T+n_V+n_A$ is the sequence length after concatenation. The terms $E^{\mathrm{pos}}, E^{\mathrm{mod}}, E^{\mathrm{data}}_k \in \mathbb{R}^{S \times d}$ denote position, modality, and dataset embeddings, respectively, with k indexing the dataset. We do not assign physical modality identities to `[CLS]` or `[TASK]`, so the corresponding rows in E^{mod} are set to zero. The coarse MSA/ER token is complemented by dataset embeddings and modality structure, which distinguish ERC and FER.

3.3 Hierarchical MoE Block

Let $X^{(\ell)} \in \mathbb{R}^{S \times d}$ denote the input to the ℓ-th Hierarchical MoE block. Each block contains three pre-normalized residual sub-layers: self-attention, hard modality experts, and task-conditioned cross experts. For $\ell = 0, \ldots, L-1$, $X^{(0)}$ is defined in Eq. 4, and the three sub-layers recursively map $X^{(\ell)}$ to $Z^{(\ell)}$, $U^{(\ell)}$, and $X^{(\ell+1)}$, respectively.

Self-attention. The first sub-layer applies self-attention [41]:

$$Z^{(\ell)} = X^{(\ell)} + \mathrm{SA}\left(\mathrm{LN}(X^{(\ell)}); M^{(\ell)}\right), \tag{5}$$

where $\mathrm{SA}(\cdot)$ is a self-attention layer, $\mathrm{LN}(\cdot)$ denotes LayerNorm [45], $M^{(\ell)}$ is the attention mask at layer ℓ, and $Z^{(\ell)} \in \mathbb{R}^{S \times d}$ is the output. In the first L_{stage} blocks, $M^{(\ell)}$ is an intra-modal mask that blocks direct attention between different modality blocks allowing global special-token communication. In later blocks, the mask is removed to allow full cross-modal fusion.

Hard modality experts. The second sub-layer contains four deterministic experts $\{E^T, E^V, E^A, E^S\}$ for text, vision, audio, and special tokens, respectively. Each token position s is dispatched according to its structural identity $m(s) \in \{T, V, A, S\}$:

$$U^{(\ell)}_s = Z^{(\ell)}_s + E^{m(s)}\left(\mathrm{LN}(Z^{(\ell)}_s)\right), \tag{6}$$

where $Z_s^{(\ell)}$ is the hidden state at position s after self-attention, and $U_s^{(\ell)}$ is the corresponding output after modality-specific refinement. The special-token expert E^S is shared by `[CLS]` and `[TASK]`, since neither token corresponds to a physical modality.

Task-conditioned cross experts. The third sub-layer contains N shared cross experts $\{E_1^{\mathrm{C}}, \ldots, E_N^{\mathrm{C}}\}$. For token position s, let $u_s = \mathrm{LN}(U_s^{(\ell)})$ and $u_{\mathrm{TASK}} = \mathrm{LN}(U_{\mathrm{TASK}}^{(\ell)})$, where $U_{\mathrm{TASK}}^{(\ell)}$ is the hidden state of the `[TASK]` token in the same block. The router predicts a task-conditioned distribution over cross experts:

$$w_s = \mathrm{softmax}\left(W_r^\top (u_s + W_\tau u_{\mathrm{TASK}})\right) \in \mathbb{R}^N, \tag{7}$$

where W_r and W_τ are learnable parameters, and w_s is the routing distribution for token s.

We select the Top-K experts and aggregate their outputs: The main setting is sparse Top-1 dispatch, with full softmax probabilities retained for straight-through router training and load balancing.

$$X_s^{(\ell+1)} = U_s^{(\ell)} + \sum_{i \in \mathcal{T}_s} \bar{w}_{s,i} E_i^{\mathrm{C}}(u_s), \tag{8}$$

where $\mathcal{T}_s = \mathrm{TopK}(w_s)$ is selected expert set for token s, $\bar{w}_{s,i}$ is normalized routing weight after Top-K selection, and $X_s^{(\ell+1)}$ is the output hidden state. Thus, each block first refines modality-specific information with hard experts and then performs task-aware interaction with routed cross experts.

3.4 Readout and Learning Objective

To handle heterogeneous label spaces, we maintain a dataset-specific prediction head ϕ_k for each dataset D_k:

$$\hat{y} = \phi_k\left(\mathrm{LN}\left(X_{\mathrm{CLS}}^{(L)}\right)\right) \in \begin{cases} \mathbb{R}, & \tau(k) = \mathrm{MSA}, \\ \mathbb{R}^{C_k}, & \tau(k) = \mathrm{ER}, \end{cases} \tag{9}$$

where $X_{\mathrm{CLS}}^{(L)}$ is the final hidden state of the `[CLS]` token, $\tau(k)$ is the task type of dataset D_k, and C_k is the number of emotion categories in D_k.

The task loss is

$$\mathcal{L}_{\mathrm{task}} = \begin{cases} |\hat{y} - y|, & \tau(k) = \mathrm{MSA}, \\ \mathrm{CE}(\hat{y}, y), & \tau(k) = \mathrm{ER}, \end{cases} \tag{10}$$

where y is the ground-truth label and $\mathrm{CE}(\cdot, \cdot)$ denotes cross-entropy loss.

To encourage balanced expert usage, we add a Switch-style auxiliary load-balancing loss [46]:

$$\mathcal{L}_{\mathrm{aux}} = N \sum_{i=1}^{N} f_i P_i, \qquad f_i = \frac{1}{K|\mathcal{B}|} \sum_{s \in \mathcal{B}} \mathbf{1}[i \in \mathcal{T}_s], \qquad P_i = \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} w_{s,i}, \tag{11}$$

where $\mathcal{B}$ is the set of routed token positions in a mini-batch, f_i is the fraction of Top-K assignments routed to expert i, P_i is average soft routing probability of expert i, and $w_{s,i}$ is the i-th entry of w_s.

The total objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{aux}} \mathcal{L}_{\mathrm{aux}}, \tag{12}$$

where λ_{aux} controls the weight of the auxiliary loss. All datasets share the same backbone and expert parameters, while prediction heads are dataset-specific. During training, each mini-batch contains samples from a single dataset so that the active head and label space are unambiguous.

3.5 Unified Checkpoint Selection

Let K_d denote the number of datasets used in unified training; in our setting, $K_d = 9$. After each epoch, we evaluate the model on the validation split of every dataset and select the checkpoint that maximizes the unified validation score:

$$\mathrm{Score}_{\mathrm{val}} = \frac{1}{K_d} \sum_{k=1}^{K_d} s_k, \qquad s_k = \begin{cases} 1 - \mathrm{MAE}_k, & \tau(k) = \mathrm{MSA}, \\ \mathrm{WA}_k, & \tau(k) = \mathrm{ER}, \end{cases} \tag{13}$$

where s_k is the validation score for dataset D_k, MAE_k is the validation mean absolute error for MSA dataset D_k, and WA_k denotes the validation weighted accuracy for ERC datasets and weighted average recall for FER datasets. The selected checkpoint is directly evaluated on all test splits without per-dataset fine-tuning or dataset-wise adaptation. All reported results therefore correspond to a single unified checkpoint for both MSA and ER. Additional parameter studies and further analysis are provided in Appendix D.

4 Experiments and Analysis

4.1 Datasets

We evaluate MEME on nine benchmarks covering MSA, ERC, and FER, where the model is jointly trained on all benchmarks and evaluated using a single composite-best checkpoint. For MSA we use four benchmarks: MOSI [15], MOSEI [9], SIMS [16], and SIMS-V2 [17]. For ER we adopt MELD [18] and IEMOCAP [19] for conversational emotion recognition, and MAFW [20], DFEW [21], and AFEW [22] for in-the-wild dynamic facial expression recognition. A full description of each dataset, our split protocol, and preprocessing pipeline are provided in Appendix C.

4.2 Baselines

We compare MEME with a diverse set of strong baselines on both MSA and ER benchmarks including task-specific models, unified multimodal models, and recent large-model-based methods. For MSA, we consider several representative task-specific methods, including TFN [47], LMF [48], MulT [3], BBFN [49], MISA [4], MMIM [50], Self-MM [51], CubeMLP [52], CENet [53], TETFN [54], TMBL [55], ALMT [56], and KuDA [57], and recent unified or cross-dataset methods, such as FDMER [58], AOBERT [59], DMD [60], ReconBoost [61], CGGM [62], DecAlign [26], MMSLF [63], UniMSE [10], UniSA [12], MSE-Adapter [13], and EGMF [14]. For ER, besides unified multimodal models, we further compare with several competitive baselines from conversational emotion recognition and dynamic facial expression recognition, including MGLRA [64], LS-DGNN+ICL [6], MMDFN [65], C3D [66], Former-DFER [67], DFER-CLIP [68], A3lign-DFER [69], CLVSR [27], R1-Omni [70], I3D-RGB [71], STT [72], NR-DFERNet [73], DPCNet [74], IAL [75], and CLIPER [11]. Due to differences in data preprocessing pipelines, experimental settings, and the availability of open-source implementations, a fully consistent reproduction of some baselines is not feasible. Therefore, we report original results to ensure comprehensive comparison.

4.3 Evaluation Metrics

For MSA, we report mean absolute error (MAE), binary accuracy (Acc-2), and F1 score (F1). Specifically, Acc-2 represents the negative/positive protocol on MOSI and MOSEI, and the positive/non-positive protocol on SIMS and SIMS-V2, aligning with MMSA framework [76]. For ER, we adopt the standard metrics of each benchmark: classification accuracy (Acc) and weighted F1 (WF1) on MELD and IEMOCAP, and weighted average recall (WAR) and unweighted average recall (UAR) on MAFW, DFEW, and AFEW. All results of MEME are averaged over three tests, and full implementation details and hyperparameters are provided in Appendix F.

4.4 Performance Comparison

Tables 1 and 2 compare MEME against task-specific, unified, and large-model baselines on all nine benchmarks. MEME uses a single composite-best checkpoint. On MSA, MEME attains the best MAE on three of four datasets, with the largest gains on the Chinese benchmarks: MAE on SIMS drops from 0.370 (MMSLF) to 0.310 (-16.2% relative), and SIMS-V2 reaches the best Acc-2 (82.40) with MAE close to KuDA's. The improvements are most pronounced in MAE, suggesting that the unified model does not merely sharpen polarity separation but also better preserves continuous sentiment intensity. On MOSI, MEME is best across all three metrics, exceeding the unified UniMSE and the recent task-specific KuDA. On MOSEI, MEME remains competitive on MAE but shows weaker calibration on boundary-sensitive samples; we revisit this trade-off in Section 4.10. On ER, the largest absolute gains appear on the FER benchmarks. WAR on MAFW improves by $+5.18\%$ over R1-Omni and UAR by $+6.61\%$ over CLVSR; on AFEW, WAR / UAR improve by $+3.88\%$ /

Table 1: Performance comparison on four MSA datasets. The best and second-best results are marked in **bold** and underline, respectively. Results marked with † are taken from [57]; results marked with ‡ are taken from [26]; all remaining baseline results are extracted from their original papers.

Method	MOSI			MOSEI			SIMS			SIMS-V2		
	MAE	Acc-2	F1	MAE	Acc-2	F1	MAE	Acc-2	F1	MAE	Acc-2	F1
TFN†	0.947	79.08	79.11	0.572	81.89	81.74	0.432	78.38	78.62	0.303	80.14	80.14
LMF†	0.950	79.18	79.15	0.576	83.48	83.36	0.441	77.77	77.88	0.367	74.18	73.88
MulT†	0.879	80.98	80.95	0.559	84.63	84.52	0.453	78.56	79.66	0.291	80.68	80.73
BBFN†	0.796	82.47	82.44	0.545	85.73	85.56	0.430	78.12	77.88	0.300	78.53	78.41
MISA†	0.776	83.54	83.58	0.557	84.67	84.66	-	-	-	-	-	-
MMIM†	0.744	84.30	84.23	0.550	85.42	85.26	-	-	-	-	-	-
Self-MM†	0.708	85.46	85.43	0.531	85.15	84.90	0.425	80.04	80.44	0.311	79.69	79.76
CubeMLP†	0.755	82.32	84.23	0.537	85.23	85.04	0.419	77.68	77.59	0.334	78.53	78.53
CENet†	-	-	-	-	-	-	0.471	77.90	77.53	0.310	79.56	79.63
TETFN†	-	-	-	-	-	-	0.420	81.18	80.24	0.310	79.73	79.81
TMBL†	-	-	-	-	-	-	0.429	79.12	78.75	0.313	80.46	80.36
ALMT†	0.712	85.82	85.86	0.530	85.99	86.05	0.408	78.77	78.71	0.308	79.59	79.51
KuDA†	0.705	86.43	86.46	0.529	86.46	86.59	0.408	80.74	80.71	**0.271**	82.11	82.04
FDMER‡	0.760	83.01	83.22	0.571	83.88	83.35	0.424	-	78.74	-	-	-
AOBERT‡	0.780	83.03	83.02	0.588	83.90	83.64	0.430	-	78.55	-	-	-
DMD‡	0.744	83.24	83.55	0.561	84.17	83.88	0.421	-	79.88	-	-	-
ReconBoost‡	0.793	82.59	82.72	0.599	82.98	83.14	0.413	-	80.41	-	-	-
CGGM‡	0.787	82.73	82.89	0.584	83.72	83.94	0.417	-	80.12	-	-	-
DecAlign‡	0.735	85.75	85.82	0.543	86.48	86.07	0.403	-	81.85	-	-	-
MMSLF	0.734	86.61	86.69	0.539	86.62	86.71	0.370	83.06	84.06	-	-	-
UniMSE	0.691	86.90	86.42	0.523	**87.50**	**87.46**	-	-	-	-	-	-
UniSA$_{\text{BART}}$	0.742	84.11	-	0.587	84.93	-	-	-	-	-	-	-
MSE (llama2-7B)	-	-	-	0.501	86.74	86.51	-	-	-	0.382	75.53	75.44
EGMF (llama2-7B)	-	-	-	**0.500**	87.16	86.97	-	-	-	0.364	77.04	76.93
MEME	**0.688**	**87.35**	**87.34**	0.524	85.39	84.77	**0.310**	**84.03**	**84.31**	0.283	**82.40**	**82.47**

$+5.62\%$ over CLIPER. These gains are notable because R1-Omni and CLVSR are dedicated dynamic FER methods built on large vision–language backbones, whereas MEME is a single shared model. On conversational ER, MEME achieves the best Acc on both MELD (66.62) and IEMOCAP (76.82, $+2.87\%$ over DecAlign), again exceeding unified baselines without any dataset-specific adaptation. Overall, a compact shared backbone with frozen encoders matches or beats methods that rely on 2–7B LLM backbones, suggesting that on heterogeneous affective data the bottleneck is not raw capacity but how shared and task-conditioned computation is organized.

4.5 Comparisons with Identical Features

To examine whether the gains depend on extractor strength, we reproduce representative baselines using the same frozen features as MEME. MEME uses one shared checkpoint across all nine datasets; this constraint is not imposed on the baselines. Holding the input representations fixed helps separate downstream modeling gains from differences in extractor strength.

Table 3 shows that MEME improves both MAE and Acc-2 on MOSI and SIMS. Compared with DecAlign, MOSI MAE decreases from 0.695 to 0.688 and Acc-2 increases from 86.57 to 87.35. On SIMS, MAE decreases from 0.358 to 0.310 and Acc-2 increases from 83.37 to 84.03 over LNLN. UniMSE remains stronger on MOSEI, consistent with the joint-training trade-off discussed in Section 4.10.

The same-feature ERC/FER results show a similar overall advantage for MEME across the reported metrics. These results indicate that the gains are not solely due to stronger encoders and support the effectiveness of the shared fusion model despite its single-checkpoint constraint. See Appendix E.1 for detailed ERC/FER results and baseline attribution.

Table 2: Performance comparison on five ER datasets. The best and second-best results are marked in **bold** and underlined, respectively. All AFEW baseline results are taken from [11]; all other baseline results are extracted from their original papers.

Method	MELD		IEMOCAP		MAFW		DFEW		AFEW	
	Acc	WF1	Acc	WF1	WAR	UAR	WAR	UAR	WAR	UAR
MMDFN	62.49	59.46	68.21	68.18	-	-	-	-	-	-
MGLRA	66.40	64.90	71.30	70.10	-	-	-	-	-	-
LSDGNN+ICL	64.67	64.90	71.30	70.10	-	-	-	-	-	-
DecAlign	-	-	73.35	73.43	-	-	-	-	-	-
UniMSE	66.46	**65.51**	70.56	70.66	-	-	-	-	-	-
$\text{UniSA}_{\text{BART}}$	65.09	62.22	64.24	64.46	-	-	-	-	-	-
MSE (llama2-7B)	65.14	63.66	-	-	-	-	-	-	-	-
EGMF (llama2-7B)	66.46	65.42	-	-	-	-	-	-	-	-
C3D	-	-	-	-	42.25	31.17	53.54	42.74	-	-
Former-DFER	-	-	-	-	43.27	31.16	65.70	53.69	50.92	47.42
DFER-CLIP	-	-	-	-	52.55	39.89	71.25	59.61	-	-
A3lign-DFER	-	-	-	-	53.24	42.07	74.20	64.09	-	-
CLVSR	-	-	-	-	52.69	42.51	71.58	**64.33**	-	-
R1-Omni	-	-	-	-	57.68	40.04	65.83	56.27	-	-
I3D-RGB	-	-	-	-	-	-	-	-	45.41	41.86
STT	-	-	-	-	-	-	-	-	54.23	49.11
NR-DFERNet	-	-	-	-	-	-	-	-	53.54	48.37
DPCNet	-	-	-	-	-	-	-	-	51.67	47.86
IAL	-	-	-	-	-	-	-	-	54.59	50.84
CLIPER	-	-	-	-	-	-	-	-	56.43	52.00
MEME	**66.62**	64.57	**76.82**	**75.87**	**62.86**	**49.12**	**75.91**	62.22	**60.31**	**57.62**

Table 3: Same-feature MSA reproductions. Values are three-run mean $\pm$ standard deviation.

Method	MOSI		MOSEI		SIMS	
	MAE	Acc-2	MAE	Acc-2	MAE	Acc-2
MulT	0.803±1.05	82.77±0.88	0.545±0.39	84.15±0.61	0.365±0.82	80.96±0.73
LNLN	0.944±1.20	79.88±0.95	0.557±0.44	84.45±0.58	0.358±0.74	83.37±0.66
UniMSE	0.743±0.52	86.13±0.54	**0.511±0.33**	**86.69±0.47**	–	–
DecAlign	0.695±0.46	86.57±0.60	0.531±0.37	85.92±0.55	0.395±0.91	81.41±0.77
MSE (Qwen-1.8B)	–	–	0.583±0.55	83.46±0.64	0.459±0.96	79.42±0.85
MEME	**0.688±0.31**	**87.35±0.62**	0.524±0.24	85.39±0.45	**0.310±0.30**	**84.03±0.58**

4.6 Ablation Study

Table 4 ablates each MEME component. Every component contributes positively. The modality expert and the task expert are the two largest sources of improvement on MSA: without the modality expert, MOSI drops by 2.29% and SIMS by 4.16%; the task expert contributes 2.75% on MOSI and 1.94% on SIMS-V2, indicating that explicit task conditioning is important for separating regression and classification objectives within a shared backbone. On ER, modality dropping is the most critical component, with removal costing 3.19% on MELD and 5.48% on AFEW, while the auxiliary balancing loss $\mathcal{L}_{\text{aux}}$ accounts for 2.35% on AFEW and 2.63% on SIMS. The dataset embedding shows a non-trivial interaction: removing E^{data} leaves SIMS unchanged at 84.03% but hurts SIMS-V2 by 0.78%, suggesting it specializes to absorb dataset-level distributional shifts rather than uniformly affecting all benchmarks. Beyond their direct performance contributions, the `[TASK]` token and modality dropping also bring qualitative benefits—routing specialization and optimization stability, respectively—which we analyze in the following two sections.

4.7 Effect of the Task Token

We measure per-token routing entropy of the cross-expert router's normalized softmax output. Without `[TASK]`, entropy stays near 100% at all layers indicating the load-balancing loss enforces batch-level

Table 4: Ablation study of MEME. For MSA datasets, we report Acc-2; for ER datasets, we report Acc on MELD/IEMOCAP and WAR on MAFW/DFEW/AFEW.

Model	MSA (Acc-2)				ER (Acc / WAR)				
	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
MEME	**87.35**	**85.39**	**84.03**	**82.40**	**66.62**	76.82	**62.86**	**75.91**	**60.31**
w/o $\mathcal{L}_{\text{aux}}$	85.82	84.59	81.40	81.33	65.85	76.70	62.53	75.05	57.96
w/o E^{mod}	86.43	85.03	82.28	81.24	65.27	77.19	61.17	75.52	59.27
w/o E^{data}	86.13	84.40	**84.03**	81.62	64.77	76.39	61.77	74.97	60.05
w/o `[TASK]`	85.98	84.56	82.06	81.43	65.58	77.19	62.64	75.22	57.96
w/o Modality Dropping	84.91	83.96	82.49	80.85	63.43	75.40	61.45	73.26	54.83
w/o Staged Attention	85.52	85.00	83.59	82.01	65.89	**77.37**	62.37	75.27	58.75
w/o Modality Expert	85.06	84.76	79.87	80.17	64.43	77.13	62.64	75.78	59.53
w/o Task Expert	84.60	84.98	82.28	80.46	65.12	77.00	62.48	75.05	57.96

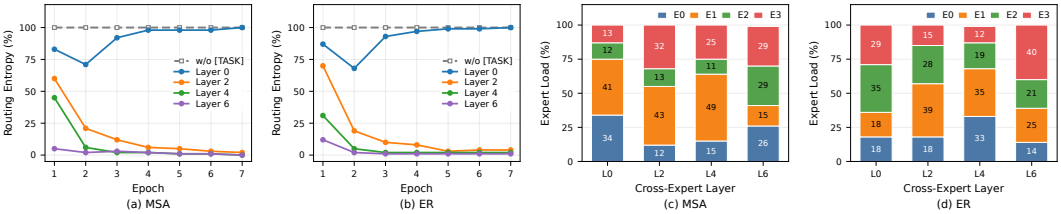


Figure 2: Cross-expert routing dynamics. (a,b) Routing entropy across layers during training, with dashed lines denoting the variant without the task token. (c,d) Expert loads for MSA and ER.

uniform expert use, but per-token distributions remain near-uniform, so the four experts receive interchangeable token populations and effectively collapse to a single averaged FFN. The `[TASK]` token breaks this symmetry—deep layers commit within five epochs while L0 stays exploratory, a natural behavior since task information needs several attention layers to propagate into other token states. The converged load profiles (Fig. 2c,d) show visibly different MSA/ER expert utilization at the same layer, confirming that the token enables task-aware specialization rather than mere load balancing. Further analysis and discussion are provided in Appendix D.3.

4.8 Effect of Modality Dropping

Beyond the consistent performance gains shown in Table 4, modality dropping also fundamentally improves optimization stability under unified training (Fig. 3). Without modality dropping, training loss converges to 0.33 but validation loss rebounds after epoch 10, reaching 0.878 when early stopping fires. With p=0.5, training loss is higher (0.77) yet validation loss descends monotonically to 0.690—below the p=0 historical best—with $\sim 4\times$ smaller seed variance. This confirms that modality dropping prevents modality-specific shortcut fitting and acts as an effective input-statistics regularizer complementary to the routing-level auxiliary loss. The analysis of different dropping probabilities in Appendix D.2 shows that p=0.5 provides the best overall trade-off.

4.9 Efficiency, Robustness and Generalization

We evaluate the accuracy–efficiency trade-off of MEME on SIMS using several model variants. Table 10 shows that the tiny variant uses only 0.22M parameters and 0.045 GFLOPs while achieving the best F1 and MAE in this efficiency comparison. This indicates that MEME can achieve superior performance with very small trainable parameter and computation budgets, demonstrating a favorable accuracy–efficiency trade-off. We also test robustness under inference-time modality dropping. Table 12 shows that MEME degrades smoothly as modality dropping increases, indicating that modality dropping and hierarchical fusion reduce over-reliance on any single modality. We further examine held-out cross-dataset and cross-task transfer in Appendix D.7, where related datasets transfer reasonably well while removing an entire task family causes a larger mismatch. Detailed results and analysis are provided in Appendices D.5, D.6, and D.7.

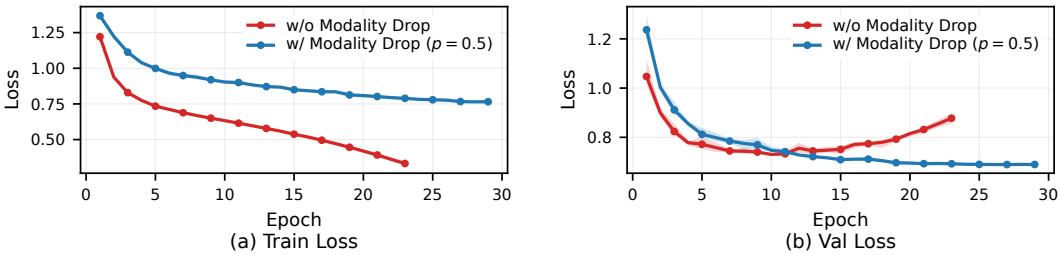


Figure 3: Loss curves under modality dropping $p \in \{0, 0.5\}$. Lines are 5-seed means; shaded regions are seed-wise min–max envelopes.

Table 5: Effect of unified training and checkpoint selection. Task-only denotes MSA-only or ER-only training. MSA results use Acc-2, ERC results use Acc, and FER results use WAR. The gap is the improvement of per-dataset checkpoint selection over the unified checkpoint in percentage points.

Setting	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
Task-only	84.76	84.95	80.31	79.50	65.04	76.08	60.20	75.27	61.10
Single checkpoint	87.35	85.39	84.03	82.40	66.62	76.82	62.86	**75.91**	60.31
Per-dataset checkpoint	**87.80**	**85.53**	**84.68**	**82.69**	**66.77**	**77.31**	**63.46**	**75.91**	**61.62**
Gap	0.45	0.14	0.65	0.29	0.15	0.49	0.60	0.00	1.31

These efficiency results describe the trainable fusion and prediction modules. For new raw inputs, frozen feature extraction is an additional cost. Appendix E.7 separates this cost from downstream inference and reports deployment-facing measurements.

4.10 Effect of Unification

To examine the effect of unification, we compare MEME with MSA-only and ER-only training, and also evaluate dataset-specific checkpoints that relax the single-checkpoint constraint. As shown in Appendix D.8, unified training improves over task-group-specific training on most datasets, suggesting that sentiment and emotion supervision provide complementary affective signals which supports our motivation. Per-dataset checkpoints further improve performance, indicating that single-checkpoint evaluation trades some per-dataset optimality for a genuinely unified model. This trade-off is expected because the composite validation score must balance heterogeneous datasets with different label spaces and task objectives. The weaker MOSEI performance and smaller unified-training gain reflect this trade-off and the remaining calibration limitation of unified MEME on weak or near-neutral sentiment samples. Detailed results and analysis are provided in Appendix D.8.

Table 5 shows gains over task-only training on eight of nine datasets, supporting complementary sentiment and emotion supervision. Per-dataset checkpoint selection uses the same jointly trained run and improves scores by at most 0.65 points except on AFEW (1.31 points). Thus, a shared checkpoint retains most performance while avoiding separate model selection for each dataset; the larger AFEW gap shows that this compromise affects the small dataset more.

5 Conclusion

In this paper, we present MEME, a lightweight hierarchical MoE framework for unified multimodal sentiment analysis and emotion recognition across MSA, ERC, and dynamic FER. Unlike prior unified affective models that mainly focus on MSA and ERC or rely on large-model adaptation, MEME targets a broader unified setting with one shared backbone and one composite-best checkpoint across nine benchmarks, without dataset-wise adaptation. By combining hard modality experts, task-conditioned cross experts, a `[TASK]` routing token, and modality dropping, MEME supports heterogeneous multimodal affective learning while maintaining favorable efficiency, robustness, and generalization. Experiments show that MEME outperforms strong task-specific baselines and recent large-model-based unified methods on most benchmarks. However, strict unification still introduces trade-offs, including some loss of per-dataset optimality and remaining calibration limitations. Future work will explore calibration-aware objectives and more adaptive balancing.

Acknowledgement

This work was supported by Shanghai Municipal Science and Technology Major Project under grant 2025SHZDZX025D05.

References

[1] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2):423–443, 2019.

[2] Soujanya Poria, Erik Cambria, Rajiv Bajpai, and Amir Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Inf. Fusion*, 37:98–125, 2017.

[3] 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 *ACL (1)*, pages 6558–6569. Association for Computational Linguistics, 2019.

[4] Devamanyu Hazarika, Roger Zimmermann, and Soujanya Poria. MISA: modality-invariant and -specific representations for multimodal sentiment analysis. In *ACM Multimedia*, pages 1122–1131. ACM, 2020.

[5] Jiawen Deng and Fuji Ren. A survey of textual emotion recognition and its challenges. *IEEE Trans. Affect. Comput.*, 14(1):49–67, 2023.

[6] Xinran Li, Xiujuan Xu, and Jiaqi Qiao. Long-short distance graph neural networks and improved curriculum learning for emotion recognition in conversation. In *ECAI*, Frontiers in Artificial Intelligence and Applications, pages 4033–4040. IOS Press, 2025.

[7] Chengyan Wu, Yiqiang Cai, Yang Liu, Pengxu Zhu, Yun Xue, Ziwei Gong, Julia Hirschberg, and Bolei Ma. Multimodal emotion recognition in conversations: A survey of methods, trends, challenges and prospects. In *EMNLP (Findings)*, pages 6257–6274. Association for Computational Linguistics, 2025.

[8] Felipe Zago Canal, Tobias Rossi Müller, Jhennifer Cristine Matias, Gustavo Gino Scotton, Antonio Reis de Sá-Junior, Eliane Pozzebon, and Antonio Carlos Sobieranski. A survey on facial emotion recognition techniques: A state-of-the-art literature review. *Inf. Sci.*, 582: 593–617, 2022.

[9] Amir Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In *ACL (1)*, pages 2236–2246. Association for Computational Linguistics, 2018.

[10] Guimin Hu, Ting-En Lin, Yi Zhao, Guangming Lu, Yuchuan Wu, and Yongbin Li. Unimse: Towards unified multimodal sentiment analysis and emotion recognition. In *EMNLP*, pages 7837–7851. Association for Computational Linguistics, 2022.

[11] Hanting Li, Hongjing Niu, Zhaoqing Zhu, and Feng Zhao. CLIPER: A unified vision-language framework for in-the-wild facial expression recognition. In *ICME*, pages 1–6. IEEE, 2024.

[12] Zaijing Li, Ting-En Lin, Yuchuan Wu, Meng Liu, Fengxiao Tang, Ming Zhao, and Yongbin Li. Unisa: Unified generative framework for sentiment analysis. In *ACM Multimedia*, pages 6132–6142. ACM, 2023.

[13] Yang Yang, Xunde Dong, and Yupeng Qiang. Mse-adapter: A lightweight plugin endowing llms with the capability to perform multimodal sentiment analysis and emotion recognition. In *AAAI*, pages 25642–25650. AAAI Press, 2025.

[14] Jiaqi Qiao, Xiujuan Xu, Xinran Li, and Yu Liu. A unified framework for emotion recognition and sentiment analysis via expert-guided multimodal fusion with large language models. *CoRR*, abs/2601.07565, 2026.

[15] Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intell. Syst.*, 31(6): 82–88, 2016.

[16] Wenmeng Yu, Hua Xu, Fanyang Meng, Yilin Zhu, Yixiao Ma, Jiele Wu, Jiyun Zou, and Kaicheng Yang. CH-SIMS: A chinese multimodal sentiment analysis dataset with fine-grained annotation of modality. In *ACL*, pages 3718–3727. Association for Computational Linguistics, 2020.

[17] Yihe Liu, Ziqi Yuan, Huisheng Mao, Zhiyun Liang, Wanqiuyue Yang, Yuanzhe Qiu, Tie Cheng, Xiaoteng Li, Hua Xu, and Kai Gao. Make acoustic and visual cues matter: CH-SIMS v2.0 dataset and av-mixup consistent module. In *ICMI*, pages 247–258. ACM, 2022.

[18] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *ACL (1)*, pages 527–536. Association for Computational Linguistics, 2019.

[19] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. IEMOCAP: interactive emotional dyadic motion capture database. *Lang. Resour. Evaluation*, 42(4):335–359, 2008.

[20] Yuanyuan Liu, Wei Dai, Chuanxu Feng, Wenbin Wang, Guanghao Yin, Jiabei Zeng, and Shiguang Shan. MAFW: A large-scale, multi-modal, compound affective database for dynamic facial expression recognition in the wild. In *ACM Multimedia*, pages 24–32. ACM, 2022.

[21] Xingxun Jiang, Yuan Zong, Wenming Zheng, Chuangao Tang, Wanchuang Xia, Cheng Lu, and Jiateng Liu. DFEW: A large-scale database for recognizing dynamic facial expressions in the wild. In *ACM Multimedia*, pages 2881–2889. ACM, 2020.

[22] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Collecting large, richly annotated facial-expression databases from movies. *IEEE Multim.*, 19(3):34–41, 2012.

[23] Saghir Alfasly, Jian Lu, Chen Xu, and Yuru Zou. Learnable irrelevant modality dropout for multimodal action recognition on modality-specific annotated videos. In *CVPR*, pages 20176–20185. IEEE, 2022.

[24] Anbin Qi, Zhongliang Liu, Xinyong Zhou, Jinba Xiao, Fengrun Zhang, Qi Gan, Ming Tao, Gaozheng Zhang, and Lu Zhang. Multimodal emotion recognition with vision-language prompting and modality dropout. In *MRAC@MM*, pages 49–53. ACM, 2024.

[25] Renjie Wu, Hu Wang, Hsiang-Ting Chen, and Gustavo Carneiro. Deep multimodal learning with missing modality: A survey. *Trans. Mach. Learn. Res.*, 2026, 2026.

[26] Chengxuan Qian, Shuo Xing, Shawn Li, Yue Zhao, and Zhengzhong Tu. Decalign: Hierarchical cross-modal alignment for decoupled multimodal representation learning. *CoRR*, abs/2503.11892, 2025.

[27] Zichen Liang, Haiying Xia, Yumei Tan, and Shuxiang Song. CLVSR: concept-guided language-visual feature learning and sample rebalance for dynamic facial expression recognition. *Cogn. Comput.*, 18(1):11, 2026.

[28] Wenhao Li, Zhibin Wu, Chong Xiao, and Qiangchang Wang. Semmsa: Latent semantic-aided robust multimodal sentiment analysis with incomplete data. *CoRR*, abs/2609.30238, 2026.

[29] Zheng Lian, Haoyu Chen, Lan Chen, Haiyang Sun, Licai Sun, Yong Ren, Zebang Cheng, Bin Liu, Rui Liu, Xiaojiang Peng, Jiangyan Yi, and Jianhua Tao. Affectgpt: A new dataset, model, and benchmark for emotion understanding with multimodal large language models. In *ICML*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[30] Siyuan Mu and Sen Lin. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *CoRR*, abs/2503.07137, 2025.

[31] Haofei Yu, Zhengyang Qi, Lawrence Jang, Russ Salakhutdinov, Louis-Philippe Morency, and Paul Pu Liang. Mmoe: Enhancing multimodal models with mixtures of multimodal interaction experts. In *EMNLP*, pages 10006–10030. Association for Computational Linguistics, 2024.

[32] Ying Cheng, Yang Li, Junjie He, and Rui Feng. Mixtures of experts for audio-visual learning. In *NeurIPS*, 2024.

[33] Jiayi Xin, Sukwon Yun, Jie Peng, Inyoung Choi, Jenna L. Ballard, Tianlong Chen, and Qi Long. I2moe: Interpretable multimodal interaction-aware mixture-of-experts. In *ICML*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[34] Yiyang Fang, Wenke Huang, Guancheng Wan, Kehua Su, and Mang Ye. EMOE: modality-specific enhanced dynamic emotion experts. In *CVPR*, pages 14314–14324. Computer Vision Foundation / IEEE, 2025.

[35] Axel Gedeon Mengara Mengara and Yeon-kug Moon. Cag-moe: Multimodal emotion recognition with cross-attention gated mixture of experts. *Mathematics*, 13(12):1907, 2025.

[36] Soumya Dutta, Smruthi Balaji, and Sriram Ganapathy. A mixture-of-experts model for multimodal emotion recognition in conversations. *Comput. Speech Lang.*, 100:101965, 2026.

[37] Ziyi Li, Wei-Long Zheng, and Bao-Liang Lu. Multimodal emotion recognition with missing modality via a unified multi-task pre-training framework. In *ACM Multimedia*, pages 5717–5725. ACM, 2025.

[38] Jinming Zhao, Ruichen Li, and Qin Jin. Missing modality imagination network for emotion recognition with uncertain missing modalities. In *ACL/IJCNLP (1)*, pages 2608–2618. Association for Computational Linguistics, 2021.

[39] Yuanzhi Wang, Yong Li, and Zhen Cui. Incomplete multimodality-diffused emotion recognition. In *NeurIPS*, 2023.

[40] Huiting Huang, Tieliang Gong, Kai He, Wen Wen, Weizhan Zhang, and Mengling Feng. Recovering coherent affective patterns: Addressing modality missing in multimodal sentiment analysis. In *AAAI*, pages 21957–21965. AAAI Press, 2026.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[42] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543, 2021.

[43] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11941–11952. IEEE, 2023.

[44] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *ICML*, Proceedings of Machine Learning Research, pages 28492–28518. PMLR, 2023.

[45] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[46] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *CoRR*, abs/2101.03961, 2021.

[47] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. In *EMNLP*, pages 1103–1114. Association for Computational Linguistics, 2017.

[48] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Efficient low-rank multimodal fusion with modality-specific factors. In *ACL (1)*, pages 2247–2256. Association for Computational Linguistics, 2018.

[49] Wei Han, Hui Chen, Alexander F. Gelbukh, Amir Zadeh, Louis-Philippe Morency, and Soujanya Poria. Bi-bimodal modality fusion for correlation-controlled multimodal sentiment analysis. In *ICMI*, pages 6–15. ACM, 2021.

[50] Wei Han, Hui Chen, and Soujanya Poria. Improving multimodal fusion with hierarchical mutual information maximization for multimodal sentiment analysis. In *EMNLP (1)*, pages 9180–9192. Association for Computational Linguistics, 2021.

[51] Wenmeng Yu, Hua Xu, Ziqi Yuan, and Jiele Wu. Learning modality-specific representations with self-supervised multi-task learning for multimodal sentiment analysis. In *AAAI*, pages 10790–10797. AAAI Press, 2021.

[52] Hao Sun, Hongyi Wang, Jiaqing Liu, Yen-Wei Chen, and Lanfen Lin. Cubemlp: An mlp-based model for multimodal sentiment analysis and depression estimation. In *ACM Multimedia*, pages 3722–3729. ACM, 2022.

[53] Di Wang, Shuai Liu, Quan Wang, Yumin Tian, Lihuo He, and Xinbo Gao. Cross-modal enhancement network for multimodal sentiment analysis. *IEEE Trans. Multim.*, 25:4909–4921, 2023.

[54] Di Wang, Xutong Guo, Yumin Tian, Jinhui Liu, Lihuo He, and Xuemei Luo. TETFN: A text enhanced transformer fusion network for multimodal sentiment analysis. *Pattern Recognit.*, 136:109259, 2023.

[55] Jiehui Huang, Jun Zhou, Zhenchao Tang, Jiaying Lin, and Calvin Yu-Chian Chen. TMBL: transformer-based multimodal binding learning model for multimodal sentiment analysis. *Knowl. Based Syst.*, 285:111346, 2024.

[56] Haoyu Zhang, Yu Wang, Guanghao Yin, Kejun Liu, Yuanyuan Liu, and Tianshu Yu. Learning language-guided adaptive hyper-modality representation for multimodal sentiment analysis. In *EMNLP*, pages 756–767. Association for Computational Linguistics, 2023.

[57] Xinyu Feng, Yuming Lin, Lihua He, You Li, Liang Chang, and Ya Zhou. Knowledge-guided dynamic modality attention fusion framework for multimodal sentiment analysis. In *EMNLP (Findings)*, Findings of ACL, pages 14755–14766. Association for Computational Linguistics, 2024.

[58] Dingkang Yang, Shuai Huang, Haopeng Kuang, Yangtao Du, and Lihua Zhang. Disentangled representation learning for multimodal emotion recognition. In *ACM Multimedia*, pages 1642–1651. ACM, 2022.

[59] Kyeonghun Kim and Sanghyun Park. AOBERT: all-modalities-in-one BERT for multimodal sentiment analysis. *Inf. Fusion*, 92:37–45, 2023.

[60] Yong Li, Yuanzhi Wang, and Zhen Cui. Decoupled multimodal distilling for emotion recognition. In *CVPR*, pages 6631–6640. IEEE, 2023.

[61] Cong Hua, Qianqian Xu, Shilong Bao, Zhiyong Yang, and Qingming Huang. Reconboost: Boosting can achieve modality reconcilement. In *ICML*, Proceedings of Machine Learning Research, pages 19573–19597. PMLR / OpenReview.net, 2024.

[62] Zirun Guo, Tao Jin, Jingyuan Chen, and Zhou Zhao. Classifier-guided gradient modulation for enhanced multimodal learning. In *NeurIPS*, 2024.

[63] Haoyu Zhang, Yinan Zhang, Chaolong Ying, Xiaoying Tang, and Tianshu Yu. Improving task-specific multimodal sentiment analysis with general MLLMs via prompting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[64] Tao Meng, Fuchen Zhang, Yuntao Shou, Hongen Shao, Wei Ai, and Keqin Li. Masked graph learning with recurrent alignment for multimodal emotion recognition in conversation. *IEEE ACM Trans. Audio Speech Lang. Process.*, 32:4298–4312, 2024.

[65] Dou Hu, Xiaolong Hou, Lingwei Wei, Lian-Xin Jiang, and Yang Mo. MM-DFN: multimodal dynamic fusion network for emotion recognition in conversations. In *ICASSP*, pages 7037–7041. IEEE, 2022.

[66] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497. IEEE Computer Society, 2015.

[67] Zhiwei Huang, Yu Zhu, Hangyu Li, and Dawei Yang. Dynamic facial expression recognition based on spatial key-points optimized region feature fusion and temporal self-attention. *Eng. Appl. Artif. Intell.*, 133:108535, 2024.

[68] Zengqun Zhao and Ioannis Patras. Prompting visual-language models for dynamic facial expression recognition. In *BMVC*, page 98. BMVA Press, 2023.

[69] Zeng Tao, Yan Wang, Junxiong Lin, Haoran Wang, Xinji Mai, Jiawen Yu, Xuan Tong, Ziheng Zhou, Shaoqi Yan, Qing Zhao, Liyuan Han, and Wenqiang Zhang. A^3lign-dfer: Pioneering comprehensive dynamic affective alignment for dynamic facial expression recognition with CLIP. *CoRR*, abs/2403.04294, 2024.

[70] Jiaxing Zhao, Xihan Wei, and Liefeng Bo. R1-omni: Explainable omni-multimodal emotion recognition with reinforcement learning. *CoRR*, abs/2503.05379, 2025.

[71] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 4724–4733. IEEE Computer Society, 2017.

[72] Fuyan Ma, Bin Sun, and Shutao Li. Spatio-temporal transformer for dynamic facial expression recognition in the wild. *CoRR*, abs/2205.04749, 2022.

[73] Hanting Li, Mingzhe Sui, Zhaoqing Zhu, and Feng Zhao. Nr-dfernet: Noise-robust network for dynamic facial expression recognition. *CoRR*, abs/2206.04975, 2022.

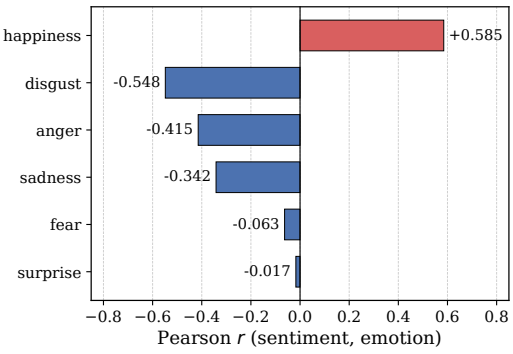
[74] Yan Wang, Yixuan Sun, Wei Song, Shuyong Gao, Yiwen Huang, Zhaoyu Chen, Weifeng Ge, and Wenqiang Zhang. Dpcnet: Dual path multi-excitation collaborative network for facial expression representation learning in videos. In *ACM Multimedia*, pages 101–110. ACM, 2022.

[75] Hanting Li, Hongjing Niu, Zhaoqing Zhu, and Feng Zhao. Intensity-aware loss for dynamic facial expression recognition in the wild. In *AAAI*, pages 67–75. AAAI Press, 2023.

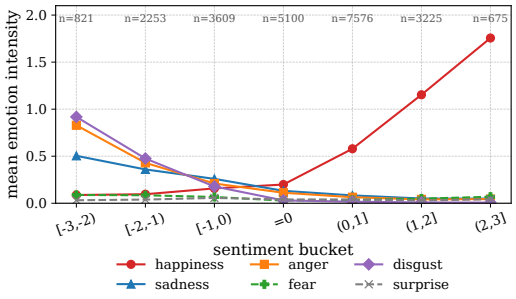
[76] Huisheng Mao, Ziqi Yuan, Hua Xu, Wenmeng Yu, Yihe Liu, and Kai Gao. M-SENA: an integrated platform for multimodal sentiment analysis. In *ACL (demo)*, pages 204–213. Association for Computational Linguistics, 2022.

[77] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *CoRR*, abs/1905.00641, 2019.

[78] Panagiotis Tzirakis, George Trigeorgis, Mihalis A. Nicolaou, Björn W. Schuller, and Stefanos Zafeiriou. End-to-end multimodal emotion recognition using deep neural networks. *IEEE J. Sel. Top. Signal Process.*, 11(8):1301–1309, 2017.



(a) Pearson correlation between sentiment labels and emotion intensities on CMU-MOSEI.



(b) Mean emotion intensity across sentiment buckets on CMU-MOSEI.

Figure 4: Relationship between sentiment labels and emotion intensities on CMU-MOSEI. The two plots show that sentiment and emotion supervision are related rather than independent, supporting unified modeling of MSA and ER.

A Limitations

MEME unifies MSA, ERC, and dynamic FER with a lightweight hierarchical MoE backbone and a strict single-checkpoint protocol, but several limitations remain. The composite-best checkpoint enables genuine unified evaluation, yet it trades off some per-dataset optimality. Sentiment calibration is still imperfect on MOSEI, where neutral and weak-sentiment samples dominate and high-intensity cases remain harder to calibrate. In addition, MEME operates on frozen text, visual, and audio features, which keeps the trainable backbone compact but makes the system dependent on the availability of pre-extracted modality features. These limitations point to future work on adaptive balancing strategies, calibration-aware objectives, and more flexible feature extraction.

B Correlation between sentiment and emotion labels on CMU-MOSEI

To provide empirical motivation for unified modeling, we analyze the relationship between sentiment and emotion labels on CMU-MOSEI [9], one of the few multimodal affective benchmarks that contains both sentiment and emotion annotations for the same utterances. As shown in Figure 4a, sentiment is positively correlated with happiness and negatively correlated with several negative emotions such as disgust, anger, and sadness. Figure 4b further shows how mean emotion intensity varies across sentiment buckets. These trends suggest that MSA and ER are not independent problems, but share part of their underlying affective structure, which motivates studying them within a unified multimodal framework.

C Datasets and Preprocessing

C.1 Dataset Descriptions

MSA datasets. CMU-MOSI [15] is an English YouTube monologue corpus with utterance-level sentiment annotations on a continuous scale of $[-3, 3]$. CMU-MOSEI [9] extends MOSI to a much larger collection of YouTube monologues from over 1,000 speakers, using the same continuous $[-3, 3]$ sentiment scale. SIMS [16] and SIMS-V2 [17] are Chinese multimodal sentiment datasets covering both monologue and conversational scenarios; both provide fine-grained sentiment labels on a continuous scale of $[-1, 1]$ together with per-modality annotations. All four MSA datasets provide text, visual, and audio inputs for each utterance, and we use the multimodal sentiment label as the regression target.

ERC datasets. MELD [18] contains multi-party dialogues from the TV series Friends, where each utterance is annotated with one of seven emotions (*anger, disgust, fear, joy, neutral, sadness, surprise*). IEMOCAP [19] consists of dyadic conversations from ten actors across five sessions; following the

widely adopted UniMSE protocol [10], we use the six-class subset (*anger, excited, frustrated, neutral, sad, happy*). Both datasets provide aligned text, visual, and audio information for each utterance.

Dynamic FER datasets. MAFW [20] is a large-scale in-the-wild dynamic facial expression dataset with 11 single-label compound emotions (*anger, disgust, fear, happiness, neutral, sadness, surprise, contempt, anxiety, helplessness, disappointment*); each clip includes the original video, audio track, and a textual description. DFEW [21] provides seven-class emotional video clips (*happiness, sadness, neutral, anger, surprise, disgust, fear*) collected from movies, with synchronized audio. AFEW [22] is also a seven-class dataset with the same emotion taxonomy as DFEW, sourced from movie clips and containing video and audio but no transcripts.

C.2 Splits and Joint-Training Protocol

Because MEME is trained as a single unified model rather than separate per-dataset models, we use the standard public test split of each benchmark whenever available and construct the remaining training/validation partitions so that one model can be selected with one composite-best criterion.

Official splits. For MOSI, MOSEI, SIMS, SIMS-V2, and MELD, we use the official train/validation/test splits released by the dataset authors without modification.

IEMOCAP, MAFW, and DFEW. These three benchmarks are commonly evaluated under 5-fold protocols, where results are averaged across the five folds. Such a protocol is not directly compatible with our single-run, single-checkpoint unified training setting. We therefore fix one held-out fold for each benchmark: Session 5 for IEMOCAP, and set_1 for MAFW and DFEW. These are standard evaluation choices in prior work, and the remaining data are split into training and validation subsets.

AFEW. Since the official test split is not publicly available, we report results on the validation set as previous work did [11].

Split statistics. Table 6 summarises the resulting numbers of samples used in our experiments.

Table 6: Dataset splits used in MEME.

Dataset	Task	#Train	#Valid	#Test	#Total
MOSI	MSA (reg.)	1,284	229	686	2,199
MOSEI	MSA (reg.)	16,326	1,871	4,659	22,856
SIMS	MSA (reg.)	1,368	456	457	2,281
SIMS-V2	MSA (reg.)	2,722	647	1,034	4,403
MELD	ERC (7-cls)	9,989	1,108	2,606	13,703
IEMOCAP	ERC (6-cls)	5,183	575	1,622	7,380
MAFW	FER (11-cls)	6,600	733	1,839	9,172
DFEW	FER (7-cls)	8,421	935	2,341	11,697
AFEW	FER (7-cls)	773	383	383	1,539
Total		**52,666**	**6,937**	**15,627**	**75,230**

C.3 Data Preprocessing and Feature Extraction

Data preprocessing. For MELD and IEMOCAP, we apply RetinaFace-based [77] face cropping before visual feature extraction to focus the visual stream on the target speaker in multi-person scenes. No additional face preprocessing is used for the other datasets.

Feature extraction. Visual frames are encoded with SigLIP-SO400M [43] at the patch-token level, yielding a 1,152-dimensional token sequence per frame. Audio is encoded with Whisper-large-v3 [44], and we use the encoder hidden states as a 1,280-dimensional sequence. Text is encoded with DeBERTa-v3-large [42] into a 1,024-dimensional token sequence. For datasets without native transcripts, for example AFEW, we use a fixed task prompt as the text input. All three encoders are kept frozen, and MEME is trained directly on the extracted multimodal features.

Table 7: Effect of model scale and MoE routing configuration. For MSA datasets we report Acc-2; for ER datasets we report Acc (MELD/IEMOCAP) or WAR (MAFW/DFEW/AFEW).

d_{model}	d_{ffn}	L	N	K	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
256	**768**	**8**	**4**	**1**	**87.35**	85.39	84.03	**82.40**	**66.62**	76.82	62.86	**75.91**	60.31
384	1024	8	4	1	84.60	84.37	84.68	80.85	65.00	75.77	59.00	74.67	56.40
512	1536	8	4	1	82.16	81.81	80.96	79.40	55.76	72.69	45.02	72.06	55.09
256	768	12	4	1	86.74	84.26	82.28	80.75	65.69	76.45	63.30	75.82	58.22
256	768	4	4	1	85.21	**85.50**	82.71	81.33	65.81	77.68	62.10	75.78	57.96
256	768	8	4	2	87.04	84.87	80.53	79.79	65.62	77.31	61.94	75.57	58.22
256	768	8	5	2	85.98	84.92	83.81	82.11	66.04	77.00	61.50	75.44	57.96
256	768	8	6	3	86.13	84.59	82.71	80.85	65.54	76.45	62.48	75.61	58.75
256	768	8	6	2	85.82	84.92	81.84	82.40	65.73	76.33	61.77	75.65	60.05
256	768	8	8	4	84.91	83.90	**84.90**	80.56	65.66	76.94	63.24	75.10	59.53
256	768	8	8	2	85.98	85.11	83.59	81.82	65.35	**77.87**	61.83	75.52	**61.88**
128	256	4	4	2	84.76	84.87	79.43	80.27	65.12	75.96	63.02	75.69	56.66
128	256	3	2	1	86.13	84.98	82.28	82.21	65.77	75.83	63.19	76.04	59.01
96	256	3	2	1	83.54	84.70	80.31	80.66	65.77	76.63	61.88	75.87	58.49
64	192	3	2	1	84.76	84.87	82.28	82.40	65.08	76.88	62.59	75.82	58.22
64	128	2	2	1	84.45	84.73	84.25	80.85	65.46	76.33	**63.46**	75.44	56.40
32	64	2	2	1	85.21	85.28	84.68	81.92	65.00	74.48	61.28	74.16	51.44
24	48	2	2	1	85.37	84.89	83.81	81.24	65.62	76.26	58.46	73.47	49.61

D Additional Experiments and Analysis

D.1 Effect of Model Scale and MoE Routing Configuration

We study how trainable capacity and MoE routing choices affect unified affective learning by varying the hidden size, FFN size, depth, number of cross experts, and Top-K value. The results in Table 7 show that the default configuration achieves the best overall balance across MSA, ERC, and FER. Increasing model size or routing capacity can improve isolated metrics, but the gains are not consistent across the nine datasets. This suggests that strict unified training is not simply limited by trainable parameter count. With frozen encoders and heterogeneous supervision, larger modules may fit dataset-specific patterns more strongly, while a compact expert structure gives more stable shared representations.

The smaller variants further show that MEME benefits from its hierarchical organization rather than parameter scale alone. Several low-capacity configurations remain competitive on both MSA and ER benchmarks, indicating that modality-specific refinement followed by task-conditioned cross-modal interaction provides a useful inductive bias. In this setting, stable specialization and sparse routing are more important than uniformly increasing depth or width, which motivates the compact configuration used in the main experiments.

D.2 Effect of Training Modality Dropping Probability

We study the training modality dropping probability to understand how incomplete-modality regularization affects unified optimization. The results in Table 8 show that removing modality dropping weakens performance on most datasets, which is consistent with the validation-loss behavior in Fig. 3. In unified training, different datasets can induce different modality shortcuts, such as stronger textual reliance in MSA and ERC or stronger visual reliance in FER. If the model always observes complete inputs, it can overfit these dataset-specific modality biases instead of learning more transferable affective evidence. Modality dropping implicitly trains the model over multiple modality subsets, making the shared backbone less dependent on any single dominant cue.

The best overall trade-off appears around $p = 0.5$. At this level, modality dropping is strong enough to regularize the shared backbone and encourage cross-modal redundancy, but not so strong that the model loses the benefit of multimodal complementarity. When p becomes too large, the model too often receives overly sparse evidence, and both MSA and ER performance start to degrade. This indicates that modality dropping in MEME is not mainly a missing-modality recovery objective, but a

Table 8: Ablation study on the modality dropping probability used during training. For MSA datasets, we report Acc-2; for ER datasets, we report Acc for MELD/IEMOCAP and WAR for MAFW/DFEW/AFEW.

p	MSA (Acc-2)				ER (Acc/WAR)				
	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
0.0	84.91	83.96	82.49	80.85	63.43	75.40	61.45	73.26	54.83
0.1	85.67	84.48	82.93	82.50	64.85	75.89	61.39	74.88	58.75
0.2	86.89	85.00	82.26	81.04	65.08	**78.24**	**63.13**	75.52	57.18
0.3	86.28	84.15	79.87	80.73	66.12	77.37	58.18	75.35	58.22
0.4	85.21	85.33	82.49	**82.69**	**66.65**	76.39	62.64	74.54	59.53
0.5	**87.35**	**85.39**	**84.03**	82.40	66.62	76.82	62.86	**75.91**	**60.31**
0.6	84.91	85.03	83.81	82.11	65.46	76.51	61.23	75.27	58.49
0.7	86.59	84.76	83.81	81.24	65.77	76.82	60.20	75.22	59.27
0.8	84.60	84.07	83.59	81.62	65.00	76.57	61.07	74.75	56.14
0.9	83.99	84.87	81.18	80.27	64.58	74.60	58.24	73.52	55.61
1.0	77.90	82.86	79.43	76.50	57.41	65.47	53.40	66.08	56.66

unified-training regularizer that improves robustness by reshaping how the model distributes affective information across modalities.

D.3 Further Analysis of the Task Token

We further analyze the task token because it plays a different role from dataset embeddings and prediction heads. Dataset embeddings identify distributional shifts among datasets, and dataset-specific heads handle different output spaces, while the task token provides a task-level conditioning signal inside the shared MoE backbone. This distinction is important in unified training, where MSA and ER share most parameters but induce different optimization pressures. MSA requires polarity- and intensity-sensitive regression, whereas ER requires categorical discrimination from conversational or facial cues. Without an explicit task anchor, the router has to infer these differences only from token states, which can be ambiguous in early training.

The routing behavior in Fig. 2 shows that the task token helps turn load balancing into meaningful expert specialization. The auxiliary loss encourages balanced expert usage, but it does not define what each expert should specialize in. Without the task token, a balanced router can still assign nearly uniform probabilities to all experts, causing different experts to receive similar token populations and converge toward interchangeable functions. With the task token, the router receives a stable task-dependent signal, so expert usage becomes both balanced and semantically differentiated.

The layer-wise pattern also explains why early layers remain more exploratory. At shallow layers, task information is still concentrated in the task-token position and has not fully propagated to modality tokens through self-attention. Deeper layers receive more task-aware token states and therefore show clearer MSA/ER routing preferences. This indicates that the task token does not merely improve performance as an additional parameter; it changes how the shared backbone resolves task conflicts by routing different affective objectives through different cross-expert pathways.

D.4 Effect of Pooled Tokens

Table 9 studies the number of learned-query pooled tokens for text, vision, and audio. The token budget controls how much modality information is retained before the shared MoE backbone. Too few tokens may discard useful temporal or semantic details, while too many tokens increase the routing and fusion burden under a fixed trainable capacity. The default setting $(n_T, n_V, n_A) = (16, 8, 8)$ gives a strong overall balance across the nine datasets. It allocates a larger budget to text, which is important for MSA and ERC, while keeping visual and audio tokens compact enough for efficient cross-modal interaction. Larger token budgets occasionally improve individual datasets but do not consistently improve the unified checkpoint, suggesting that learned-query pooling is most effective when it compresses modality evidence rather than simply preserving more tokens.

Table 9: Effect of the number of pooled modality tokens.

(n_T, n_V, n_A)	MSA (Acc-2)				ER (Acc/WAR)				
	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
$(4, 2, 2)$	85.82	84.81	81.84	81.53	65.50	76.51	**63.57**	75.27	59.27
$(8, 4, 4)$	87.04	84.89	80.74	79.79	65.62	77.31	61.99	75.57	58.22
$(8, 4, 8)$	87.04	85.17	81.18	81.43	65.85	**77.50**	**63.57**	75.69	59.27
$(8, 8, 4)$	87.04	84.56	82.06	81.14	65.96	76.08	61.83	75.74	58.22
$(8, 8, 8)$	86.13	84.84	82.28	81.33	64.77	**77.50**	61.99	76.04	**60.57**
$(12, 6, 6)$	83.84	85.06	**84.25**	**82.88**	65.12	75.65	61.28	**76.33**	58.49
$(16, 4, 4)$	85.82	84.70	82.93	81.14	65.92	76.39	62.75	74.93	**60.57**
$(\mathbf{16}, \mathbf{8}, \mathbf{8})$	**87.35**	**85.39**	84.03	82.40	**66.62**	76.82	62.86	75.91	60.31
$(16, 16, 8)$	86.13	84.92	83.15	82.11	65.35	76.02	62.37	75.05	59.79
$(16, 16, 16)$	85.98	84.31	81.62	81.24	64.89	75.77	62.26	75.27	57.96
$(24, 12, 12)$	86.13	84.59	83.59	82.50	65.73	76.26	61.61	75.95	58.49
$(32, 16, 16)$	85.52	84.98	83.59	81.62	66.31	76.39	62.53	75.48	59.53

D.5 Efficiency Analysis

We evaluate the accuracy–efficiency trade-off of MEME variants on SIMS, with the corresponding configurations summarized in Table 11. Since all reproduced methods are evaluated on frozen extracted features, the comparison in Table 10 focuses on the trainable multimodal fusion and prediction modules rather than the common feature extractors. MEME-Base achieves strong performance with moderate computation, while MEME-Small substantially reduces parameters and FLOPs with only a limited performance drop. MEME-Tiny further reduces the trainable module to 0.22M parameters and 0.045 GFLOPs while achieving the best F1 and MAE on SIMS. The fact that MEME-Tiny outperforms the larger MEME-Base on SIMS is partly related to the scale of the dataset. Since SIMS is smaller than large-scale benchmarks such as MOSEI, increasing the complexity of the trainable fusion module does not necessarily improve generalization and can instead make the model more sensitive to dataset-specific patterns. With strong frozen encoders, a small hierarchical MoE can already extract sufficient affective evidence from compact pooled tokens, while a larger fusion module may introduce unnecessary capacity for this dataset. This explains why the smallest variant can outperform larger MEME variants in this efficiency study.

MEME is also computationally efficient because its MoE routing activates only a subset of cross experts for each token. Compared with dense execution of all experts, the effective computation of the cross-expert layer scales with the selected Top-K experts rather than the total number of experts. Under the same total expert parameterization, sparse dispatch or optimized expert kernels can further reduce practical GFLOPs because only the selected experts are executed for each token. Together with learned-query pooling, which compresses long modality sequences before fusion, this makes MEME efficient in both parameter count and conditional computation.

D.6 Robustness Analysis

Tables 12 and 13 evaluate robustness under inference-time modality dropping. This setting measures whether a model can still make reliable predictions when some modality signals are unavailable or corrupted. MEME obtains strong averaged results across MSA, ERC, and FER, especially on MSA metrics and several ER benchmarks. The detailed drop-rate results further show a gradual degradation pattern as the inference-time dropping probability increases, rather than an abrupt collapse. This indicates that the model learns partially redundant affective evidence across modalities and does not depend entirely on one dominant input source.

The robustness behavior is closely related to the training design. Modality dropping exposes the model to incomplete modality combinations during training, while hard modality experts allow the available modality tokens to be refined independently before fusion. Task-conditioned cross experts then integrate whichever modality evidence remains. This structure is well aligned with heterogeneous affective data, where text, audio, and visual cues may vary in reliability across datasets and task families. Therefore, the robustness results are not an isolated missing-modality experiment, but a consequence of the same design that stabilizes unified training.

Table 10: Efficiency and performance comparison on the SIMS dataset. Results marked with † are from [63]; all other results are reproduced from open-source implementations using the same pre-extracted multimodal features. Since all reproduced models are evaluated directly on frozen extracted features, the reported parameter counts exclude the common feature extractors, which contain approximately 1.5B parameters in total.

Method	Parameters	GFLOPs	Inference Time	F1	MAE
TFN†	35.63M	0.101	3.46s	77.83	0.434
MISA†	21.66M	7.33	12.32s	76.54	0.451
Self-MM†	0.38M	6.66	11.40s	77.72	0.418
TETFN†	1.53M	6.72	26.57s	79.34	0.422
ALMT†	2.60M	7.00	16.08s	80.17	0.421
MMSLF†	2.54M	96.16	12.31s	84.06	0.370
MulT	1.17M	0.623	415.821s	84.54	0.370
LNLN	7.36M	0.867	6.675s	83.33	0.373
DecAlign	4.49M	0.345	181.970s	70.76	0.352
MSE(Qwen-1.8B)	1.54M + 1.8B	131.642	60.52s	59.01	0.607
MEME-Base	42.60M	1.373	8.475s	84.31	0.310
MEME-Small	2.79M	0.239	3.472s	82.76	**0.302**
MEME-Tiny	**0.22M**	**0.045**	**3.345s**	**84.97**	**0.302**

Table 11: Configurations of MEME variants used in the efficiency analysis. d_{model} denotes the hidden dimension, d_{ffn} denotes the FFN hidden dimension, L denotes the number of Hierarchical MoE blocks, (n_T, n_V, n_A) denotes the number of pooled text, vision, and audio tokens, N denotes the number of cross experts, and K denotes the Top-K routing value.

Model	d_{model}	d_{ffn}	L	(n_T, n_V, n_A)	N	K
MEME-Base	256	768	8	$(16, 8, 8)$	4	1
MEME-Small	128	256	3	$(8, 4, 4)$	2	1
MEME-Tiny	32	64	2	$(8, 4, 4)$	2	1

D.7 Cross-Dataset and Cross-Task Generalization Analysis

We further examine whether MEME learns transferable affective representations beyond the standard nine-dataset joint-training setting. In each experiment, one dataset or one task family is excluded from unified training, and the held-out data are evaluated directly using the closest available proxy head. Specifically, AFEW and MAFW are evaluated with the DFEW head, SIMS with the SIMS-V2 head, and IEMOCAP with the MELD head. For cross-family transfer, ERC datasets are evaluated with the DFEW head when all ERC data are excluded, and FER datasets are evaluated with the MELD head when all FER data are excluded. When label spaces differ, we use the predefined label mapping and evaluate the mapped label space only.

The leave-one-dataset results in Table 14 show that MEME transfers well when the held-out dataset has a related training source. SIMS remains close to the fully unified result when evaluated through the SIMS-V2 head, and AFEW also retains strong performance when transferred from DFEW. MAFW shows a larger drop, which is expected because its label space and in-the-wild distribution are less aligned with DFEW after mapping. IEMOCAP is more difficult to transfer from MELD, reflecting the stronger role of speaker interaction, dialogue style, and dataset-specific emotion distribution in ERC. The setting that excludes AFEW, IEMOCAP, MAFW, and SIMS simultaneously still gives non-trivial performance on all four held-out datasets, suggesting that the shared backbone captures reusable affective structure rather than merely memorizing individual benchmarks.

The leave-task-family results are substantially weaker, especially when all FER datasets are excluded and facial emotion datasets must be evaluated through a conversational ER head. This drop is important for interpreting the main unified results. It shows that the strong performance of MEME in the standard setting should not be attributed only to generic pretrained features or proxy-head transfer. Joint supervision from MSA, ERC, and FER is necessary for learning a unified affective model that performs well across all nine benchmarks. In other words, these generalization experiments support

Table 12: Averaged robustness comparison under random modality dropping at inference time. Best results are in **bold** and second-best results are underlined. All results of baselines are reproduced from open-source implementations using the same pre-extracted multimodal features. All results are averaged over three runs using different random seeds to reduce the effect of stochasticity from modality dropping.

Method	MOSI		MOSEI		SIMS		MELD		IEMOCAP		MAFW		DFEW	
	MAE	Acc-2	MAE	Acc-2	MAE	Acc-2	Acc	WF1	Acc	WF1	WAR	UAR	WAR	UAR
MulT	1.069	72.22	0.646	78.91	0.516	70.67	-	-	-	-	-	-	-	-
LNLN	1.113	71.05	0.649	78.37	0.442	78.41	-	-	-	-	–	-	-	-
UniMSE	1.102	70.33	0.650	60.17	-	-	37.69	39.39	47.38	46.97	-	-	-	-
DecAlign	1.459	44.53	0.727	75.72	0.503	64.60	-	-	44.91	39.16	-	-	-	-
MM-DFN	-	-	-	-	-	-	49.60	47.72	**51.98**	**51.90**	-	-	-	-
MSE(Qwen-1.8B)	-	-	0.676	56.39	0.609	62.68	-	-	38.32	39.03	-	-	-	-
R1-Omni	-	-	-	-	-	-	-	-	-	-	36.94	26.96	55.27	**44.48**
DFER-CLIP	-	-	-	-	-	-	-	-	-	-	32.65	23.00	53.20	42.24
MEME	**0.906**	**78.05**	**0.612**	**80.21**	**0.404**	**78.48**	**58.23**	**52.14**	48.79	43.87	**37.11**	**29.18**	**55.53**	44.17

Table 13: Detailed robustness results of MEME under different modality drop probabilities. The last row reports the average over all 11 drop probabilities.

Drop Prob.	MOSI		MOSEI		SIMS		SIMS-V2		MELD		IEMOCAP		MAFW		DFEW	
	MAE	Acc-2	MAE	Acc-2	MAE	Acc-2	MAE	Acc-2	Acc	WF1	Acc	WF1	WAR	UAR	WAR	UAR
0.0	0.688	87.35	0.524	85.39	0.310	84.03	0.283	82.40	66.62	64.57	76.82	75.87	62.86	49.12	75.91	62.22
0.1	0.728	85.92	0.542	83.76	0.333	83.15	0.297	81.82	64.70	62.10	69.40	67.96	55.75	43.76	71.18	58.30
0.2	0.771	83.69	0.561	82.67	0.351	81.25	0.312	81.14	62.47	59.00	62.56	60.90	50.24	38.89	66.25	53.81
0.3	0.829	82.11	0.580	81.94	0.380	79.87	0.332	79.50	60.55	56.26	56.33	54.28	44.50	35.17	61.27	49.70
0.4	0.871	79.57	0.600	80.94	0.404	78.92	0.341	78.56	58.67	53.55	50.14	47.21	38.59	30.86	57.48	46.37
0.5	0.928	77.69	0.620	80.13	0.413	78.70	0.359	77.01	56.95	50.59	45.09	41.06	33.30	26.59	52.84	42.54
0.6	0.985	74.80	0.637	78.74	0.427	77.24	0.372	76.76	56.02	48.82	40.51	35.31	29.49	23.20	49.86	38.86
0.7	1.010	73.02	0.655	78.24	0.433	77.02	0.386	74.95	54.22	46.10	37.30	30.48	26.23	20.71	46.21	35.91
0.8	1.041	72.46	0.666	77.07	0.458	74.47	0.400	73.50	54.04	45.25	33.97	25.16	23.49	18.39	44.21	33.71
0.9	1.051	71.54	0.674	77.02	0.470	75.20	0.408	72.82	53.11	43.77	32.24	22.28	22.20	17.32	42.59	32.16
1.0	1.063	70.43	0.677	76.45	0.470	73.45	0.399	73.92	53.22	43.66	32.33	22.10	21.53	16.98	43.07	32.26
Avg.	**0.906**	**78.05**	**0.612**	**80.21**	**0.404**	**78.48**	**0.354**	**77.49**	**58.23**	**52.14**	**48.79**	**43.87**	**37.11**	**29.18**	**55.53**	**44.17**

the value of unified training while also showing its boundary. Although removing an entire task family creates a larger mismatch in modality dependence, label semantics, and decision boundaries, MEME retains meaningful held-out performance on unseen datasets and under cross-family transfer, suggesting favorable generalization beyond the standard joint-training setting.

D.8 Effect of Unification

We analyze the effect of unified training by comparing task-group-specific training, the standard single-checkpoint MEME, and a relaxed per-dataset checkpoint setting. The results in Table 15 show that unified training improves over MSA-only and ER-only training on most datasets, indicating that sentiment and emotion supervision provide complementary affective signals. MSA contributes polarity- and intensity-sensitive supervision, while ER contributes categorical emotion cues from conversational and facial contexts. The comparison with MEME* further shows the cost of strict unification. Per-dataset checkpoint selection improves several metrics, indicating that relaxing the single-checkpoint constraint and allowing per-dataset adaptation can further improve performance. This is expected because different datasets have distinct label distributions, modality dependencies, and validation dynamics. Therefore, the composite-best checkpoint trades some per-dataset optimality for a genuinely unified model.

The remaining weakness on MOSEI is closely related to sentiment calibration. As shown in Fig. 5, MOSEI has the largest neutral region among the four MSA datasets, with 41.8% of samples falling into the neutral band. It also has the smallest proportion of strong-sentiment samples, with only

Table 14: Cross-dataset and cross-task generalization results. The "Excl. 4 datasets" setting excludes AFEW, IEMOCAP, MAFW, and SIMS simultaneously, which combines held-out samples from MSA, ERC, and FER.

Setting	MSA (Acc-2)				ER (Acc / WAR)				
	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
MEME	87.35	85.39	84.03	82.40	66.62	76.82	62.86	75.91	60.31
Excl. AFEW	-	-	-	-	-	-	-	-	58.22
Excl. SIMS	-	-	83.59	-	-	-	-	-	-
Excl. MAFW	-	-	-	-	-	-	58.67	-	-
Excl. IEMOCAP	-	-	-	-	-	59.02	-	-	-
Excl. 4 datasets	-	-	81.84	-	-	56.90	59.26	-	54.57
Excl. all ERC	-	-	-	-	42.75	61.04	-	-	-
Excl. all FER	-	-	-	-	-	-	43.61	33.82	42.82

Table 15: Effect of unified training. Best results are in **bold** and second-best results are underlined. MEME* denotes the per-dataset best checkpoint setting relaxing the single-checkpoint constraint.

Setting	MSA (Acc-2)				ER (Acc / WAR)				
	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
MSA-only	84.76	84.95	80.31	79.50	–	–	–	–	–
ER-only	–	–	–	–	65.04	76.08	60.20	75.27	61.10
MEME	87.35	85.39	84.03	82.40	66.62	76.82	62.86	**75.91**	60.31
MEME*	**87.80**	**85.53**	**84.68**	**82.69**	**66.77**	**77.31**	**63.46**	**75.91**	**61.62**

about 20.3% in the two strong bands combined. This distribution makes MOSEI more sensitive to calibration around weak or near-neutral sentiment. Under unified training, the objective is influenced by the dense weak-sentiment region, which helps the main binary metrics but makes high-intensity sentiment separation more difficult.

Table 16 supports this interpretation. On MOSI, SIMS, and SIMS-V2, the Non0 improvement is larger than the Has0 improvement, suggesting that unified training improves non-neutral sentiment discrimination. On MOSEI, however, the Non0 gain is only $+0.44$, smaller than the Has0 gain of $+1.33$, and the Strong subset decreases by 0.48. This pattern is consistent with the label distribution in Fig. 5: when weak or neutral sentiment dominates, the unified model becomes better calibrated for the majority region but less effective on the minority strong-sentiment subset. Therefore, the weaker MOSEI performance in the main table reflects a remaining calibration limitation under imbalanced sentiment intensity, rather than contradicting the overall benefit of unified training.

D.9 Effect of the Auxiliary Load-Balancing Weight

We study the effect of the auxiliary load-balancing weight λ_{aux}, which controls the strength of the expert-usage regularization in the task-conditioned cross-expert router. The results in Table 17 show that the default value $\lambda_{\text{aux}} = 0.05$ gives the best overall balance across the nine datasets. A very small weight weakens the load-balancing constraint and makes expert utilization less stable, while an overly large weight can over-regularize the router and interfere with task-driven specialization. This is especially important in unified training, where the router must serve heterogeneous objectives rather than simply distribute tokens uniformly. The trend suggests that $\mathcal{L}_{\text{aux}}$ is most useful when it acts as a soft regularizer instead of a dominant optimization target. Smaller values such as 0.01 and 0.03 still produce competitive results on several datasets, but they are less consistent across MSA and ER. Larger values such as 0.1 and 0.15 can improve isolated metrics, for example on DFEW, but they reduce the overall balance and hurt several sentiment or emotion benchmarks. Therefore, $\lambda_{\text{aux}} = 0.05$ is used in the main experiments because it provides sufficient expert coverage while preserving task-conditioned routing specialization.

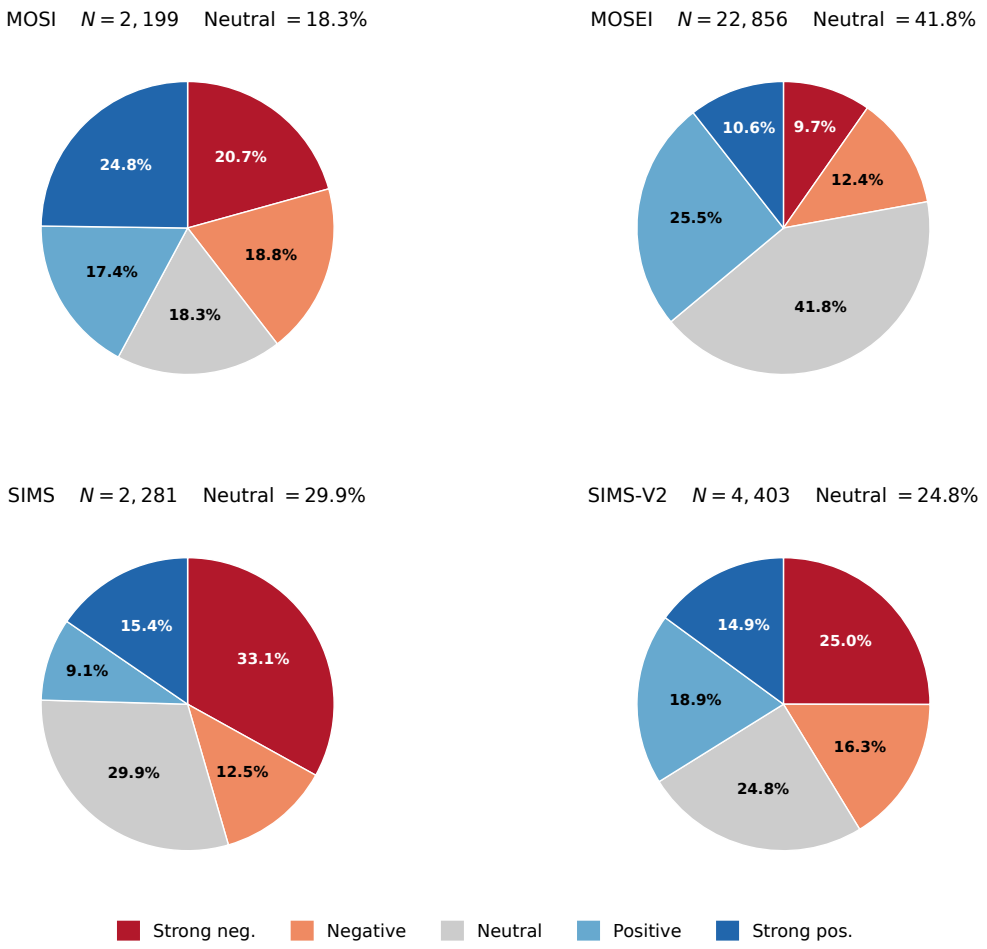


Figure 5: Sentiment label distribution of the four MSA benchmarks.

Table 16: Sentiment calibration analysis on MSA datasets. Has0 and Non0 denote Acc-2 computed with and without zero-labeled samples, respectively. Strong denotes Acc-2 on high-intensity sentiment samples, using $|y| > 0.5$ for MOSI and MOSEI, and $|y| > 0.2$ for SIMS and SIMS-V2. The last row reports the improvement of MEME over the MSA-only setting in percentage points.

Setting	MOSI			MOSEI			SIMS			SIMS-V2		
	Has0	Non0	Strong	Has0	Non0	Strong	Has0	Non0	Strong	Has0	Non0	Strong
MSA-only	83.24	84.76	88.10	79.52	84.95	**89.06**	80.31	81.70	84.69	79.50	81.87	85.73
MEME	**85.28**	**87.35**	**89.83**	**80.85**	**85.39**	88.58	**84.03**	**86.08**	**88.75**	**82.40**	**85.16**	**89.44**
Δ	+2.04	+2.59	+1.73	+1.33	+0.44	-0.48	+3.72	+4.38	+4.06	+2.90	+3.29	+3.71

D.10 Effect of Staged Attention

We further study the number of early blocks using intra-modal attention masks, denoted by L_{stage}. When $L_{\text{stage}} = 0$, all layers allow full cross-modal attention from the beginning. When L_{stage} is larger, modality-specific attention is preserved for more layers before full fusion is enabled. The results in Table 18 show that the default setting $L_{\text{stage}} = 2$ achieves the strongest overall performance across the unified benchmark suite.

The effect of staged attention reflects a balance between modality refinement and cross-modal fusion. Without staged attention, the model mixes modalities too early, which can introduce noisy interactions before the modality experts have sufficiently refined text, visual, and audio representations. In contrast, using too many staged layers delays cross-modal communication and can weaken the multimodal

Table 17: Effect of the auxiliary load-balancing weight.

λ_{aux}	MSA (Acc-2)				ER (Acc / WAR)				
	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
0.01	86.13	84.81	83.59	81.53	65.39	77.44	62.75	75.95	57.44
0.03	85.82	84.20	82.93	81.04	65.35	**77.56**	62.26	75.82	59.79
0.05	**87.35**	**85.39**	**84.03**	**82.40**	**66.62**	76.82	**62.86**	75.91	**60.31**
0.1	86.28	85.00	81.40	81.24	65.00	76.57	61.39	75.14	56.92
0.15	84.91	84.87	83.15	81.04	65.43	76.39	61.77	**76.46**	58.75

Table 18: Effect of the number of staged-attention blocks.

L_{stage}	MSA (Acc-2)				ER (Acc / WAR)				
	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
0	85.52	**85.39**	82.06	80.37	65.89	**77.37**	62.37	75.27	58.75
2	**87.35**	**85.39**	**84.03**	**82.40**	**66.62**	76.82	**62.86**	**75.91**	60.31
4	85.82	84.81	82.93	80.27	65.85	77.13	61.12	75.61	**60.84**
8	84.60	84.81	82.71	79.98	66.27	77.00	61.66	75.48	57.96

interaction needed for affective prediction. The default setting keeps the first two blocks more modality-aware and then allows full fusion in later layers, which aligns with the hierarchical design of MEME: lower layers emphasize modality-specific refinement, while higher layers perform task-conditioned multimodal interaction. Although longer or shorter staging can improve isolated datasets, $L_{\text{stage}} = 2$ provides the most stable unified performance.

E Additional Controlled Comparisons and Deployment Analysis

E.1 Baseline Attribution and Same-Feature Reproductions

The baseline entries in Tables 1 and 2 are published results under their respective original feature and preprocessing settings, with secondary sources identified in the captions. The MSA reproductions in Table 3 and ERC/FER reproductions in Table 19 use the same frozen features as MEME. MEME uses one shared checkpoint across all nine datasets, while the baselines are not required to share a checkpoint across datasets. The text, visual, and audio features are extracted by DeBERTa-v3-large, SigLIP-SO400M, and Whisper-large-v3, respectively, with preprocessing as in Appendix C. This comparison controls the input representations without equating the models' training and checkpoint-selection procedures. EGMF has no public code; MMML uses a different extractor per dataset; MEME's per-task baselines used different extractors. We therefore retain the original-paper results for breadth and provide same-feature reproductions for attribution.

On ERC, MEME improves MELD Acc from 65.35 for UniMSE to 66.62 and IEMOCAP Acc from 75.86 for MSE-Adapter to 76.82. On FER, WAR improves from 49.87 for DFER-CLIP to 62.86 on MAFW and from 68.62 for R1-Omni to 75.91 on DFEW. MEME also exceeds the highest reproduced UAR values on both datasets, reaching 49.12 versus 43.82 on MAFW and 62.22 versus 58.91 on DFEW. It therefore leads on all eight reported ERC/FER metrics and on twelve of fourteen metrics across MSA, ERC, and FER, with UniMSE stronger on both MOSEI metrics. These results indicate that stronger frozen encoders alone do not account for the gains. The simple fusion controls below further examine the architectural contribution, while the study in Appendix E.7 tests dataset-provided SIMS features.

E.2 Simple and Dense Fusion Controls

Concat + MLP uses neither attention nor MoE, single cross-attention adds generic attention fusion, and the shared-FFN control removes experts and routing. Unlike the external baseline reproductions above, these architectural controls use MEME's unified training and single-checkpoint protocol with the same frozen features. MEME has the highest mean result on all five metrics in Table 20, exceeding

Table 19: Same-feature ERC (upper panel) and FER (lower panel) reproductions. Values are three-run mean ± standard deviation.

	MELD		IEMOCAP	
Method	Acc	WF1	Acc	WF1
UniMSE	65.35±0.62	63.99±0.70	75.83±0.68	74.83±0.66
DecAlign	–	–	74.45±0.72	74.61±0.69
MM-DFN	62.72±0.75	61.47±0.80	75.03±0.71	72.23±0.78
MSE (Qwen-1.8B)	–	–	75.86±0.70	75.31±0.67
MEME	**66.62±0.71**	**64.57±0.76**	**76.82±0.84**	**75.87±0.77**
	MAFW		DFEW	
Method	WAR	UAR	WAR	UAR
R1-Omni	48.40±0.90	43.82±1.05	68.62±0.82	58.91±0.88
DFER-CLIP	49.87±0.86	40.72±1.10	64.31±0.79	53.84±0.95
MEME	**62.86±0.91**	**49.12±1.15**	**75.91±0.83**	**62.22±0.94**

Table 20: Simple/dense controls and expert ablations, three-run mean ± standard deviation. MSA uses Acc-2; MELD uses Acc; FER uses WAR.

Model	MOSI	SIMS	MELD	MAFW	DFEW
Concat + MLP	82.90±0.86	78.92±0.97	63.12±0.78	60.88±0.95	73.64±0.88
Single cross-attention	83.84±0.82	80.15±0.90	63.74±0.74	61.35±0.91	74.22±0.83
Shared FFN	84.52±0.76	81.35±0.84	64.15±0.69	61.83±0.87	74.71±0.80
w/o Modality Expert	85.06±0.81	79.87±1.05	64.43±0.73	62.64±0.91	75.78±0.78
w/o Task Expert	84.60±0.79	82.28±0.82	65.12±0.68	62.48±0.89	75.05±0.77
MEME	**87.35±0.62**	**84.03±0.58**	**66.62±0.71**	**62.86±0.91**	**75.91±0.83**

the shared-FFN control by 2.83 points on MOSI, 2.68 on SIMS, 2.47 on MELD, 1.03 on MAFW, and 1.20 on DFEW. The modality-expert ablation has its largest effects on SIMS, while removing task experts lowers performance across several task families. Together, these comparisons support complementary modality-specific refinement and task-conditioned interaction beyond generic fusion. Differences on FER should be interpreted in light of the three-run variability.

E.3 Run Variability and Statistical Comparisons

Table 21 complements the main-table means with three-run standard deviations. Table 22 reports representative two-tailed t-tests from the controlled comparisons. Each reported p-value is below the conventional 0.05 threshold, supporting the corresponding improvement for these comparisons. These are individual, unadjusted tests based on three runs, not evidence that every small main-table difference is significant. In particular, the controlled MOSEI results favor UniMSE, and no uniform superiority claim is made.

E.4 Sensitivity of Unified Checkpoint Selection

Equation 13 gives each benchmark equal weight, although the metrics have different scales. To examine the effect of this choice, we compare the original criterion with per-dataset min–max and z-score normalization, $\exp(-\mathrm{MAE})$ for MSA, ordinal ranks, and a geometric mean of min–max scores. All six validation criteria select epoch 16. Across epochs 13–19, mean test goodness ranges from 0.572 to 0.583, indicating limited sensitivity to the tested checkpoint choices. The comparison with per-dataset checkpoints in Table 5 shows that the remaining trade-off is larger on the small AFEW dataset.

E.5 Sparse Routing and Task Granularity

The main configuration uses sparse Top-K routing with $K = 1$. The selected weight is then one in the forward pass. This is intentional hard dispatch, distinct from collapse in which the same

Table 21: Variability of MEME main results over three runs.

Dataset	Metric 1		Metric 2	
MOSI	MAE	0.688±0.31	Acc-2	87.35±0.62
MOSEI	MAE	0.524±0.24	Acc-2	85.39±0.45
SIMS	MAE	0.310±0.30	Acc-2	84.03±0.58
SIMS-V2	MAE	0.283±0.26	Acc-2	82.40±0.49
MELD	Acc	66.62±0.71	WF1	64.57±0.76
IEMOCAP	Acc	76.82±0.84	WF1	75.87±0.77
MAFW	WAR	62.86±0.91	UAR	49.12±1.15
DFEW	WAR	75.91±0.83	UAR	62.22±0.94
AFEW	WAR	60.31±1.32	UAR	57.62±1.41

Table 22: Representative two-tailed tests in the same-feature setting.

Dataset	Comparison	Metric	p-value
MOSI	MEME vs. DecAlign	MAE	0.018
MOSI	MEME vs. DecAlign	Acc-2	0.041
SIMS	MEME vs. LNLN	MAE	0.006
SIMS	MEME vs. LNLN	Acc-2	0.048
MELD	MEME vs. UniMSE	Acc	0.031
DFEW	MEME vs. R1-Omni	WAR	0.022

expert receives nearly all tokens or specialization fails. The implementation computes full softmax probabilities w_s and uses the straight-through estimator

$$\widetilde{w}_s = \mathrm{stopgrad}(\widehat{w}_s - w_s) + w_s, \tag{14}$$

where $\widehat{w}_s$ is the sparse, renormalized Top-K distribution. The full probabilities also enter the load-balancing loss together with hard expert-selection fractions. They therefore remain useful for training and for the routing-entropy analysis. Entropy is measured before sparse dispatch: near-uniform soft probabilities indicate weak specialization, not a dense average of all expert outputs in a Top-1 forward pass.

The K sweep is already reported in Table 7. Larger K sometimes helps individual datasets, but does not consistently improve the shared-checkpoint result. The computational benefit of Top-K also depends on execution: the reference implementation evaluates an expert over the full token tensor if that expert is selected anywhere in the batch. Sparse token dispatch or optimized expert kernels would be needed to obtain ideal per-token compute scaling. Measured serving time should therefore be distinguished from ideal active-expert FLOPs.

The two-valued `[TASK]` token is a coarse routing anchor, not the only source of task information. Dataset embeddings are added to all tokens, modality embeddings and hard experts preserve text/visual/audio structure, and output heads define the applicable label spaces. These signals allow ERC and FER to share ER supervision while retaining dataset and modality distinctions. Table 4 shows that removing task tokens, dataset embeddings, or modality experts affects different subsets of benchmarks; no single signal accounts for all specialization. Pilot runs with separate ERC/FER tokens did not produce a consistent gain beyond dataset embeddings, so the main configuration retains MSA/ER tokens. These pilots are not a comprehensive task-granularity study.

ERC predictions use the current utterance's multimodal features. MEME does not add explicit dialogue-history or speaker-identity modules. Its ERC results should be interpreted as utterance-level evaluation, and do not establish that conversational context is unnecessary.

E.6 Closed-Set Automatic Head Selection

Unification refers to one shared backbone and checkpoint, with small dataset-specific read-outs for incompatible targets. All prediction heads together contain 0.30M parameters, less than 0.8% of the trainable model. In an application with a known output space, that space determines the head and dataset embedding. To evaluate an interface without dataset identity at test time, we train a one-hidden-layer classifier on frozen input features. Its targets are the training examples' benchmark/head

Table 23: Predicted versus oracle head selection. The change is measured on $(1 - \mathrm{MAE}) \times 100$ for MSA and Acc/WAR in percentage points for ER; negative indicates degradation.

	MOSI	MOSEI	SIMS	SIMS-V2	MELD	IEMOCAP	MAFW	DFEW	AFEW
Head accuracy (%)	99.7	100.0	3.9	96.7	99.9	100.0	99.3	100.0	100.0
Metric change	-0.15	0.00	-0.26	-0.62	-0.04	0.00	-0.60	0.00	0.00

Table 24: SIMS dataset-provided-feature comparison. Inference time covers 457 test examples. Parameters and GFLOPs exclude feature preparation; MSE includes its frozen Qwen backbone in execution. Means ± standard deviations use three runs except the single-run MSE row.

Method	Trainable	Frozen LLM	GFLOPs	Time (s)	F1	MAE
MulT	1.11M	–	0.582	25.34	76.52±0.85	0.441±0.86
LNLN	7.11M	–	0.722	4.80	**78.57±0.34**	0.443±0.41
DecAlign	4.41M	–	0.228	18.04	78.22±0.47	0.440±0.55
MSE (Qwen-1.8B)	1.54M	1.8B	131.230	11.71	61.15	0.557
MEME-Tiny	**0.22M**	–	**0.010**	**2.45**	77.13±0.85	**0.423±0.30**

identities; the dataset identifier is a supervision label, not an input feature. No MEME gradients are used. At inference, the predicted identity selects both the dataset embedding and prediction head.

The router reaches 99.78% task-family accuracy and 96.87% overall head accuracy. Table 23 reports per-dataset behavior. The low SIMS self-routing accuracy mainly reflects confusion with SIMS-V2, which shares Chinese-language sentiment inputs and the same continuous label range. Accordingly, head identity accuracy and task prediction quality need not change proportionally. The mean end-to-end loss is 0.19 points, with a maximum loss of 0.62 points. The overall head accuracy is sample-weighted; it is not the unweighted average of the per-dataset accuracies.

This is closed-set routing among nine supported heads. It supports known sentiment and emotion applications, but does not provide unseen-task recognition, unknown-label-space inference, or automatic adaptation to new emotion taxonomies.

E.7 Efficiency Boundaries and Deployment-Facing Measurements

Trainable versus deployed parameters. The main results use MEME-Base with 42.6M trainable parameters, including its heads; the original SIMS profile reports 1.373 GFLOPs for the downstream module (Table 10). These counts exclude the frozen extractors. If all three extractors are deployed for new raw inputs, their approximately 1.5B parameters raise the full pipeline to approximately 1.543B parameters. For cached-feature serving, only the downstream module needs to be resident in the prediction process. Thus trainable-module size and total raw-input deployed size describe different boundaries. An optional automatic head classifier is a separate module and is not included in these serving measurements.

Approximate extraction costs are 0.6 GFLOPs per text token, 0.075 TFLOPs per audio second, and 0.67 TFLOPs per video frame under a $2 \times$ parameters $\times$ tokens estimate. They are indicative operation counts rather than measured end-to-end latency. Actual input cost depends on tokenization, audio duration, resolution, and especially the configurable frame count. Frozen extraction can be performed offline for cached pipelines, but remains part of inference on new raw inputs. Methods using exactly the same frozen features share that extraction stage; methods with other encoders do not necessarily have the same extraction cost.

Dataset-provided SIMS features. Table 24 uses the simple features distributed with SIMS, without applying the large frozen extractors used in the main setting. Here, "raw features" means dataset-provided feature tensors, not unprocessed video or audio. Feature preparation is outside the measured boundary. MEME-Tiny attains the best MAE, uses 0.22M trainable parameters and 0.010 GFLOPs, and has the shortest measured test time; its F1 is 1.44 points below LNLN. This supports an efficient accuracy–cost trade-off even with simpler input features. The corresponding strong-feature MEME-Tiny profile in Table 10 is a separate setting (0.045 GFLOPs), and is retained unchanged.

Table 25: Deployment-facing measurements on SIMS. Deployed parameters count all model parameters resident in the reported prediction stage, excluding prior feature preparation. Throughput and per-sample time are amortized across all 457 examples. MSE generation throughput and per-sample time are derived from its reported total time.

Prediction setting	Trainable	Deployed	GFLOPs	Batch	ms/sample	Samples/s	Total (s)
MEME-Base	42.6M	42.6M	0.905	64	0.409	2444.5	0.187
MEME-Tiny	0.22M	0.22M	0.010	1	5.36	186.7	2.45
MSE single forward	1.54M	≈1.802B	131.23	1	25.63	39.0	11.71
MSE generate	1.54M	≈1.802B	–	1	75.34	13.3	34.43

Deployment-facing serving test. We measure prediction on the actual 457-example SIMS test set with a trained checkpoint loaded once. All rows use PyTorch/CUDA on the same NVIDIA A40 GPU (48 GB), with an AMD EPYC 7513 CPU. Table 25 separates trainable parameters from all model parameters resident in each reported prediction stage. MEME consumes cached tensors; MSE-Adapter includes Qwen execution. Common hardware alone does not make these computational boundaries or batch sizes identical. The table is a downstream deployment-facing benchmark, not a complete raw-input, service/network, or batch-matched latency evaluation.

For MEME-Base, feature tensors are loaded onto the GPU before timing. CUDA events measure the forward passes after warm-up, excluding feature extraction, file I/O, and CPU–GPU transfers. Over ten repeated passes, the total time is 186.948 ± 4.258 ms. The 457 examples form seven full batches and one batch of nine, giving 23.368 ms per batch and an amortized 0.409 ms per example. The latter measures throughput rather than the response time of a single request. The serving checkpoint has a 0.905-GFLOP profile, which is reported separately from the 1.373-GFLOP configuration in Table 10.

MSE-Adapter's generate mode performs autoregressive decoding, repeatedly invoking the frozen Qwen backbone. Its single-forward mode invokes Qwen once and avoids repeated decoding. MEME has lower reported downstream computation and test time in these measurements, but MEME-Base uses batching and cached features, whereas MSE runs at batch size one with its backbone included. The timing difference therefore reflects both the model and the measurement setting, not architecture alone. Raw-input deployment must additionally account for feature extraction.

E.8 AFEW Fixed-Prompt Diagnostic

For AFEW without transcripts, the fixed text prompt cannot distinguish individual examples, although it may act as a constant anchor. Zeroing its features at inference on an available checkpoint changes WAR from 55.09 to 54.83, UAR from 50.79 to 50.64, and WF1 from 51.77 to 51.71. This diagnostic uses one available checkpoint rather than retraining an audio–visual model, and does not establish equivalence between the two training settings.

E.9 Audiovisual Emotion Recognition

Earlier audiovisual emotion recognition studies investigated affective features, temporal modeling, and feature- and decision-level fusion [**?**]. Deep audiovisual models have also learned multimodal representations and temporal fusion jointly [78]. MEME builds on this line of multimodal affective learning but addresses a different setting. It applies modality-specific refinement and task-conditioned cross-expert interaction to frozen deep features, and jointly trains one backbone across MSA, ERC, and dynamic FER with a shared checkpoint. Its focus is thus the organization of these components for unified affective learning rather than a new low-level fusion operator.

F Implementation Details and Hyperparameters

We implement MEME in PyTorch 2.4.0 with CUDA 13.0. All experiments are conducted on a Linux server equipped with NVIDIA A40 GPUs (48 GB) and AMD EPYC 7513 processors (2.6 GHz). A single model is jointly trained on all nine benchmarks listed in Appendix C and evaluated using one shared composite-best checkpoint, without any per-dataset finetuning or dataset-wise adaptation. Optimization uses AdamW with cosine learning-rate decay and linear warmup,

and model selection follows the unified validation criterion described in Section 3.5. The main hyperparameter configuration is summarized in Table 26.

Table 26: Main hyperparameter configuration of MEME.

Parameter	Value
Model	
Hidden dimension d_{model}	256
FFN hidden dimension d_{ffn}	768
Number of Hierarchical MoE blocks L	8
Text / vision / audio tokens	16 / 8 / 8
Cross experts N	4
Top-k routing	1
Staged intra-modal layers L_{stage}	2
FFN type	SwiGLU
Dropout	0.1
Modality drop probability	0.5
Auxiliary load-balance coefficient λ_{aux}	0.05
Optimization	
Optimizer	AdamW
Initial learning rate	3×10^{-4}
Weight decay	0.15
Learning-rate schedule	Cosine decay with 5% linear warmup
Gradient clipping	1.0
Batch size	64
Max epochs	30
Random seeds	1111-1113
Early-stopping patience	5
Environment	
Framework	PyTorch 2.4.0
CUDA	13.0
GPU	NVIDIA A40 (48 GB)
CPU	AMD EPYC 7513 (2.6 GHz)
Frozen feature encoders	
Text encoder	DeBERTa-v3-large (1024-d)
Vision encoder	SigLIP-SO400M (1152-d)
Audio encoder	Whisper-large-v3 (1280-d)

Impact Statement

Unified multimodal affective understanding may be useful for human-centered applications such as assistive interaction, education, and adaptive interfaces. At the same time, sentiment and emotion recognition may raise privacy and fairness concerns, especially when applied to facial, vocal, or conversational data. Such models could also be misused for emotion profiling or inappropriate automated assessment. Therefore, practical deployment should consider consent, transparency, and human oversight. Benchmark labels reflect dataset-specific annotation conventions and distributions, and aggregate accuracy does not establish demographic fairness. Predictions should not be over-interpreted as definitive measurements of an individual's emotions.

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: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations can be found in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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 data preprocessing and implementation details in Appendix C and Appendix F respectively. We also provide the code in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: All the datasets are obtained from open-source libraries. The code is available in the supplemental material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental settings and details are provided in Section 4, Appendix C and Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

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: [Yes]

   Justification: The main results are averaged over three tests and all results in Table 12 are averaged over three runs using different random seeds to reduce the effect of stochasticity from modality dropping. In addition, Fig. 3 reports 5-seed mean loss curves with min–max shaded regions to show the variance caused by modality dropping.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: [Yes]

   Justification: Appendix F reports the implementation environment. We also report model parameters, GFLOPs, and inference time on SIMS in Table 10, which provides additional information about the computational cost of the proposed model and baselines.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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: We make sure that the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We describe this in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: Properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We will release the code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.