STREAM: SAMPLE-ADAPTIVE TWO-LEVEL ROUTING EXPERTS WITH ALIGNED MAMBA FOR MULTIMODAL REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Sentiment Analysis (MSA) aims to understand human emotions by integrating acoustic, visual, and textual signals. However, existing approaches often rely on static fusion architectures with limited interpretability, overlooking the dynamic and hierarchical nature of cross-modal interactions. To address these limitations, we propose STREAM, an interpretable dynamic fusion framework for MSA. STREAM introduces an input-conditional operator routing mechanism combined with a Partial Information Decomposition (PID)-inspired Mixture-of-Experts design, enabling sample-specific modeling of unique, redundant, and synergistic cross-modal patterns. Furthermore, STREAM incorporates hierarchical representation alignment and a Collaborative Bi-Mamba module to effectively capture interactions across different semantic levels and temporal dependencies. Beyond prediction, STREAM provides interpretable insights by analyzing routing behaviors, expert activation patterns, and modality-specific contributions. We evaluate STREAM on three benchmark datasets, including CMU-MOSI, CMU-MOSEI, and CH-SIMSv2. Experimental results demonstrate competitive performance across these benchmarks, achieving state-of-the-art results on CH-SIMSv2 and strong results on CMU-MOSEI, with an Acc-7 of 54.58%, Acc-2 of 85.74%, MAE of 0.531, and Corr of 0.772. By offering both adaptive fusion and interpretable decision analysis, STREAM provides a promising foundation for future affective computing applications, including intelligent human-computer interaction and emotion-aware systems.

1 INTRODUCTION

Multimodal Sentiment Analysis (MSA) aims to infer human sentiment from acoustic, visual, and textual signals Baltrušaitis et al. (2019); Zadeh et al. (2018c). Existing fusion methods, from early fusion to attention-based transformers Tsai et al. (2019a), have achieved strong performance, but they are often limited by static architectures that cannot adapt to diverse cross-modal interaction patterns. Recent advances in multimodal representation learning and pre-trained multimodal models further demonstrate the importance of flexible cross-modal interaction modeling Rahman et al. (2020b;a). In practice, multimodal cues may convey complementary, overlapping, or emergent information. Conventional fusion pipelines typically aggregate heterogeneous modalities through a single coarse fusion stage. Such a design imposes a static topology on all samples, models cross-modal interactions in a flat manner, and obscures which modality interactions or computational pathways contribute to the final prediction.

Recent studies have drawn on Partial Information Decomposition (PID) Williams & Beer (2010); Wollstadt et al. (2023) to characterize multimodal information as unique, redundant, and synergistic components. This view is appealing for MSA, where modalities often interact through modality-specific evidence, shared consensus, and emergent cross-modal effects. Prior work has used PID-inspired ideas to analyze representations and design contrastive objectives Liang et al. (2023; 2024). However, incorporating such principles into end-to-end routing remains difficult, and most existing fusion methods still rely on static, hand-crafted topologies, in contrast to the architecture search literature that learns topologies from data Zoph & Le (2016); Liu et al. (2018a).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these issues, we propose STREAM, a hierarchical fusion framework for MSA. STREAM separately models low-level and high-level interactions, allows different semantic levels to prefer different operators and experts, and uses cross-level alignment followed by Collaborative Bi-Mamba fusion to build a shared representation. By moving beyond a single static fusion stage, STREAM aims to adaptively route diverse multimodal interaction patterns while improving the interpretability of the fusion process.

The main contributions of our work can be summarized as follows:

- We propose STREAM, a hierarchical multimodal sentiment analysis framework that integrates dual-level routing, cross-level alignment, and Collaborative Bi-Mamba fusion to capture complex multimodal interactions across semantic granularities.
- We introduce the Automated Partial-Information Mixture-of-Experts (APMoE), a module that combines input-conditional operator routing over a differentiable search space with PID-inspired expert specialization, so as to model unique, redundant, and synergistic cross-modal interactions while exposing its routing behavior through expert activations and operator weights.
- Experiments on CMU-MOSI, CMU-MOSEI, and CH-SIMSv2 demonstrate that STREAM achieves competitive or state-of-the-art performance over strong baselines, and our ablations isolate the contribution of hierarchical routing, PID-inspired specialization, cross-level alignment, and collaborative temporal fusion.

2 Related work

2.1 Multimodal fusion and alignment

Early MSA methods mainly explored feature fusion through tensor-based interactions, low-rank factorization, and memory mechanisms Zadeh et al. (2017); Liu et al. (2018b); Zadeh et al. (2018a). Transformer-based approaches further improved cross-modal alignment through attention mechanisms Tsai et al. (2019b). However, most existing methods still rely on manually designed fusion structures, limiting their adaptability to sample-specific multimodal interactions.

2.2 Representation decoupling and dynamic routing

Recent works investigate modality-specific and shared representations through factorization and mutual-information-based objectives, such as MISA, MMIM, and Self-MM Hazarika et al. (2020); Han et al. (2021); Yu et al. (2021). Meanwhile, MoE-based methods introduce adaptive expert selection for multimodal fusion Shazeer et al. (2017); Fedus et al. (2022). However, existing approaches generally employ black-box gating without explicitly modeling different information types.

2.3 Conditional computation and its relation to architecture search

Our operator router is closely related to two distinct lines of work, and we position it explicitly to avoid conflation. Differentiable architecture search Liu et al. (2018a), following reinforcement-learning-based search Zoph & Le (2016), optimizes a set of *globally shared* architecture parameters over a supernet, then discretizes them into a single fixed architecture that is retrained from scratch. Conditional computation, in contrast, keeps the full candidate set at inference time and selects among candidates *per input*; sparsely gated MoE layers Shazeer et al. (2017); Fedus et al. (2022) are the canonical instance.

Unlike conventional differentiable architecture search, which learns a single fixed topology, STREAM performs input-conditional operator routing and preserves sample-specific computation paths during inference. Therefore, our router belongs to conditional computation rather than architecture discovery.

3 Method

3.1 Unimodal feature extraction and hierarchical stream construction

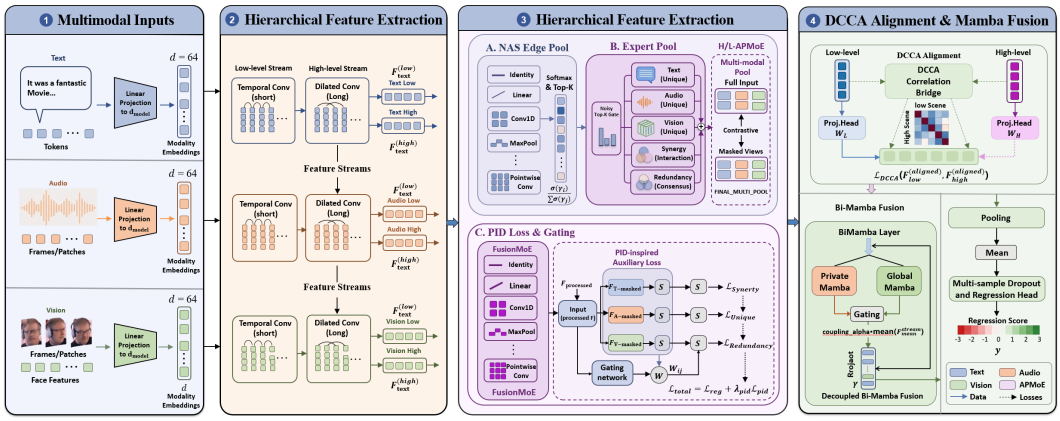


Figure 1: The overall architecture of the proposed STREAM framework. The pipeline consists of three main stages: (1) Unimodal Feature Extraction (Sec. 3.1), (2) Semantic Decoupling & Consensus via dual-level APMoE (H-APMoE and L-APMoE) and DCCA alignment (Sec. 3.2), and (3) Collaborative Fusion & Prediction utilizing the Collaborative Bi-Mamba module to output continuous sentiment intensity (Sec. 3.3).

Following common practice in MSA, we operate on pre-extracted unimodal features rather than training encoders end-to-end from raw signals.

A central design choice of STREAM is to construct two semantic levels for each modality before routing. Specifically, each modality is passed through a shallow convolutional stack, where the concatenated outputs from the earlier layers are used as the low-level representation, while the outputs from the deeper layers form the high-level representation with progressively larger receptive fields and more abstract features. We denote the resulting low- and high-level streams for modality m by $\mathbf{X}_m^{\mathrm{LL}}$ and $\mathbf{X}_m^{\mathrm{HL}}$, respectively.

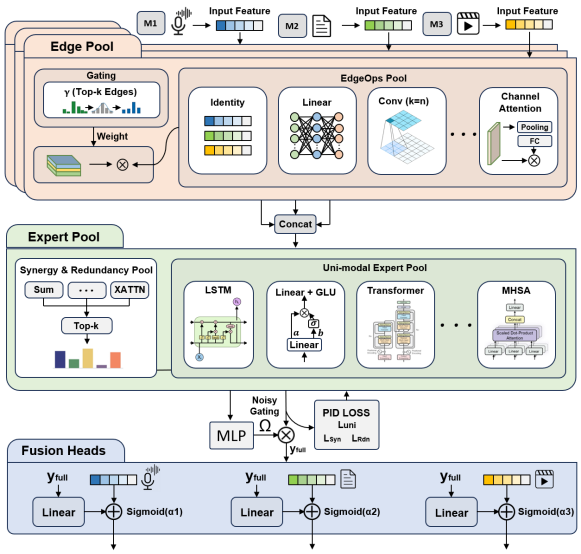


Figure 2: Internal structure of the Automated Partial-Information Mixture-of-Experts (APMoE). Inputs are dynamically routed through an edge pool and an expert pool under PID-inspired supervision and noisy gating.

3.2 APMoE

Existing multimodal routing methods commonly rely on fixed computational topologies with limited ability to adapt their operations to heterogeneous cross-modal interactions. To address this limitation, APMoE, illustrated in Figure 2, combines a differentiable operator search space with PID-inspired expert routing.

3.2.1 Input-conditional operator routing

As shown in the top panel of Figure 2, APMoE maintains two coupled search spaces: a modality-wise edge-operation pool containing transformations such as standard convolutions, dilated convolutions, and identity mappings, followed by an expert-node pool containing operators such as multi-head self-attention, depthwise convolution, and long short-term memory (LSTM). Rather than using globally fixed architecture parameters, an operator router predicts operator logits from the current input representation. For modality m and edge e, the sparse operator weights are computed as

$$\boldsymbol{\alpha}_{m,e}(\mathbf{x}) = \mathrm{TopKSoftmax}\left(g_{m,e}^{\mathrm{op}}(\mathbf{x})\right), \tag{1}$$

where $\mathbf{x}$ is the current input representation, $g^{\text{op}}_{m,e}$ denotes the input-dependent operator router, and K is the number of active operations retained by $\mathrm{TopKSoftmax}$. Only the selected operations are aggregated, and the same Top-K routing rule is used during both training and inference. As discussed in Sec. 2.3, this makes the mechanism input-conditional routing rather than architecture search: no fixed topology is derived. This operator router is distinct from the subsequent expert gate: the former selects transformations from the edge pool, whereas the latter assigns the transformed features to PID-inspired experts.

We adopt a straight-through relaxation for Top-K routing. During training, gradients are propagated through the soft distribution while only selected operators are activated in the forward pass.

3.2.2 PID-INSPIRED EXPERT ROUTING

The searched features are routed to three expert types: *Uniqueness Experts* for modality-specific information (one per modality, i.e., the Text, Audio, and Vision Experts), *Synergy Experts* for emergent cross-modal interactions, and *Redundancy Experts* for shared information.

A noisy gating network predicts input-dependent soft weights for aggregating the interaction experts. Unlike the sparse Top-K operator router above, this expert gate remains dense so that all interaction types can contribute to each sample. Following Shazeer et al. (2017), the gate perturbs its logits with learnable-scale Gaussian noise during training to encourage exploration:

$$\mathbf{w}(\mathbf{x}) = \mathrm{softmax}(\mathbf{W}_g\mathbf{x} + \boldsymbol{\epsilon} \odot \mathrm{softplus}(\mathbf{W}_n\mathbf{x}))\,, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

where $\mathbf{W}_g$ and $\mathbf{W}_n$ denote the gating and noise-scale projections, respectively. The noise is used only during training and is disabled at inference.

Sparsely gated MoE layers may suffer from expert collapse Shazeer et al. (2017). In STREAM, the PID-inspired objectives in Sec. 3.4 impose distinct functional targets on the uniqueness, synergy, and redundancy experts, which discourages routing collapse and promotes expert specialization.

Because strict PID is intractable for high-dimensional continuous representations, we adopt a perturbation-based contrastive surrogate. Under modality masking, redundant representations are encouraged to remain consistent, whereas synergistic representations are encouraged to change substantially.

L-APMoE and H-APMoE produce the low-level and high-level representations, denoted by $\mathbf{X}_{\mathrm{LL}}$ and $\mathbf{X}_{\mathrm{HL}}$, respectively. Since the two streams capture complementary semantic granularities, direct fusion may introduce semantic mismatch between representations. We therefore align them with DCCA before temporal fusion. DCCA models their nonlinear correlation, strengthening shared semantics while preserving level-specific information.

3.3 COLLABORATIVE BI-MAMBA FUSION

After decoupling and alignment, the remaining challenge is temporal modeling over long multimodal sequences. Transformer-based fusion is effective but incurs quadratic complexity. Motivated by the Mamba state space model Gu & Dao (2023) and recent Mamba-based multimodal sequence models Li et al. (2026); He et al. (2025); Li et al. (2025); Han et al. (2026), we introduce Collaborative Bi-Mamba, which combines a global temporal pathway, a private temporal pathway, and a learnable gating mechanism.

For each semantic level $i \in \{\mathrm{LL}, \mathrm{HL}\}$, we first apply layer normalization to $\mathbf{X}_i$. The normalized feature is then processed by a global Mamba branch and a private Mamba branch, producing $\mathbf{H}^{(i)}_{\text{global}}$ and $\mathbf{H}^{(i)}_{\text{priv}}$, respectively. The global branch shares parameters across the two levels, whereas the private branch is level-specific. Instead of directly adding the two branches, we use a learnable gate to adaptively fuse them at each time step:

$$\mathbf{g}_i = \sigma\big(\mathrm{Linear}([\mathbf{H}^{(i)}_{\text{global}} \,\|\, \mathbf{H}^{(i)}_{\text{priv}}])\big), \qquad \tilde{\mathbf{H}}_i = \mathbf{g}_i \odot \mathbf{H}^{(i)}_{\text{global}} + (\mathbf{1} - \mathbf{g}_i) \odot \mathbf{H}^{(i)}_{\text{priv}} \tag{3}$$

where $\sigma(\cdot)$ is the element-wise sigmoid function, $\|$ denotes channel-wise concatenation, $\odot$ denotes the Hadamard product, and $\mathbf{1}$ is an all-ones matrix of the same shape as $\mathbf{g}_i$.

To further capture reverse temporal dependencies, we additionally apply a backward private Mamba branch, whose output is denoted by $\mathbf{H}_{\text{rev}}^{(i)}$, and combine it with the gated output. The final representation is computed as

$$\mathbf{X}_i^{\text{out}} = \text{Linear}_i(\tilde{\mathbf{H}}_i + \mathbf{H}_{\text{rev}}^{(i)}) + \mathbf{X}_i, \tag{4}$$

where the additive term $\mathbf{X}_i$ forms a residual connection.

3.4 Optimization objective

We train STREAM with a joint objective consisting of a regression loss $\mathcal{L}_{\text{reg}}$, a PID-inspired decoupling loss $\mathcal{L}_{\text{pid}}$, and a cross-level alignment loss $\mathcal{L}_{\text{dcca}}$:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reg}} + \lambda_{\text{pid}}\mathcal{L}_{\text{pid}} + \lambda_{\text{dcca}}\mathcal{L}_{\text{dcca}}, \tag{5}$$

where λ_{pid} and λ_{dcca} are scaling hyperparameters.

Because strict PID is intractable in this setting, we use perturbation-based contrastive objectives as a practical surrogate. These losses do not provide strict information-theoretic guarantees, but they encourage the experts to specialize toward unique, shared, and emergent interaction patterns under modality masking. The three constraints are independently applied to the outputs of their corresponding expert groups rather than a shared representation space. Specifically, $\mathbf{Y}^u$, $\mathbf{Y}^s$, and $\mathbf{Y}^r$ denote the outputs of the Uniqueness, Synergy, and Redundancy Experts, respectively. For each expert group, $\mathbf{Y}_{\text{full}}^{\bullet}$ denotes the output obtained with all modalities available, while $\mathbf{Y}_{m_i}^{\bullet}$ denotes the output obtained by preserving only modality i and masking the remaining modalities with Gaussian noise. We design three explicit constraints:

Uniqueness constraint: To encourage each Uniqueness Expert to capture modality-specific information, the output of the expert assigned to modality i should be closer to the perturbed representation that preserves modality i than to any perturbed representation that preserves a different modality. We formulate this via a triplet margin loss, as shown in Eq. equation 6:

$$\mathcal{L}_{\text{uni}} = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j\neq i}}^{M} \max\big(0,\ d_{\cos}(\mathbf{Y}_{\text{full}}^{u,i}, \mathbf{Y}_{m_i}^{u,i}) - d_{\cos}(\mathbf{Y}_{\text{full}}^{u,i}, \mathbf{Y}_{m_j}^{u,i}) + \gamma\big), \tag{6}$$

where $\mathbf{Y}^{u,i}$ is the output of the Uniqueness Expert assigned to modality i, $d_{\cos}(\cdot,\cdot) = 1 - \cos(\cdot,\cdot)$ is the cosine distance, and $\gamma > 0$ is a predefined margin.

Synergy constraint: Synergistic information emerges from joint interactions among modalities. Thus, corrupting any modality should substantially alter the synergistic representation. We enforce this by minimizing the cosine similarity between the intact and perturbed outputs of the Synergy Experts, encouraging their separation (Eq. equation 7):

$$\mathcal{L}_{\text{syn}} = \frac{1}{M} \sum_{i=1}^{M} \cos(\mathbf{Y}_{\text{full}}^{s}, \mathbf{Y}_{m_i}^{s}). \tag{7}$$

Redundancy constraint: Redundant information represents shared consensus across modalities. Therefore, the extracted representations should remain consistent when different modalities are masked. We minimize the pairwise cosine distance between the perturbed outputs of the Redundancy Experts, as computed in Eq. equation 8:

$$\mathcal{L}_{\text{red}} = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \big(1 - \cos(\mathbf{Y}_{m_i}^{r}, \mathbf{Y}_{m_j}^{r})\big). \tag{8}$$

The total decoupling loss is $\mathcal{L}_{\text{pid}} = \mathcal{L}_{\text{uni}} + \mathcal{L}_{\text{syn}} + \mathcal{L}_{\text{red}}$. All three terms are normalized by their number of summands so that they remain on comparable scales and can be balanced by the single coefficient λ_{pid}.

3.4.1 DCCA alignment loss

To align the low-level and high-level streams, we employ Deep Canonical Correlation Analysis (DCCA). Specifically, two learnable projection heads map the matched representations $\mathbf{X}_{\text{LL}}$ and

$\mathbf{X}_{\mathrm{HL}}$ into latent spaces $\mathbf{Z}_{\mathrm{L}}, \mathbf{Z}_{\mathrm{H}} \in \mathbb{R}^{N \times d}$, where N denotes the number of aligned token pairs and d is the projection dimension. Each pair corresponds to the low-level and high-level features from the same sample and timestep.

Following standard DCCA, we first center the projected features and compute the regularized covariance matrices between the two views. The covariance matrices are then whitened to obtain the canonical correlation representation $\mathbf{T}$, whose singular values characterize the correlation between the two semantic levels. The DCCA loss is defined as the negative sum of the largest d' singular values of $\mathbf{T}$, where $d' \leq d$ denotes the number of retained canonical directions. Minimizing this loss therefore maximizes the canonical correlation between the low-level and high-level representations and encourages the two streams to capture aligned shared semantic factors.

3.4.2 Regression loss

For sentiment regression, we use the Huber loss instead of the mean squared error (MSE) loss for better robustness to outliers. We further adopt a multi-sample dropout head and average the predictions of S dropout branches during training, where S is distinct from the number of active operations K in Eq. equation 1.

4 Experiments

Method	CMU-MOSI					CMU-MOSEI				
	Acc-7↑	Acc-5↑	Acc-2↑	MAE↓	Corr↑	Acc-7↑	Acc-5↑	Acc-2↑	MAE↓	Corr↑
TFN Zadeh et al. (2017)	34.90	39.39	80.08	0.901	0.698	50.20	53.10	82.50	0.593	0.700
LMF Liu et al. (2018b)	33.20	38.13	82.50	0.917	0.695	48.00	52.90	82.00	0.623	0.673
MFN Zadeh et al. (2018a)	35.83	40.47	78.87	0.927	0.670	51.34	52.76	82.85	0.718	0.575
Graph-MFN Zadeh et al. (2018b)	34.64	38.63	78.35	0.956	0.649	51.37	52.69	83.43	0.575	0.713
MulT Tsai et al. (2019b)	40.00	42.68	83.00	0.871	0.698	51.80	54.18	82.50	0.572	0.703
MMIM Han et al. (2021)	45.80	-	84.60	0.717	**0.800**	50.10	-	83.60	0.580	**0.772**
Self-MM Yu et al. (2021)	-	-	85.98	0.713	0.798	-	-	85.17	0.540	0.765
MISA Hazarika et al. (2020)	41.37	47.08	83.58	0.777	0.778	52.05	53.63	84.67	0.558	0.752
DMD Li et al. (2023)	46.06	-	83.23	0.752	-	52.78	-	84.62	0.543	-
DDSE Jiang et al. (2025)	47.52	55.39	**86.74**	**0.699**	0.794	53.81	55.80	85.47	0.537	0.767
AGFN Wu et al. (2025)	**48.69**	-	82.75	0.710	-	54.30	-	84.01	0.535	-
EMOE Fang et al. (2025)	47.70	-	85.40	0.710	-	54.10	-	85.30	0.536	-
STREAM	47.91	**56.15**	86.32	0.711	0.790	**54.58**	**56.10**	**85.74**	**0.531**	**0.772**

Table 1: Comparison of STREAM with state-of-the-art baselines on the CMU-MOSI and CMU-MOSEI datasets. The best results in each column are highlighted in **bold**; the second-best results are underlined. "-" indicates that the value was not reported in the original paper.

4.1 Datasets and metrics

We evaluate STREAM on three widely used MSA benchmarks: CMU-MOSI Zadeh et al. (2016), CMU-MOSEI Zadeh et al. (2018c), and CH-SIMSv2 Liu et al. (2022). CMU-MOSI and CMU-MOSEI are English multimodal sentiment datasets annotated with continuous sentiment scores in $[-3, 3]$, while CH-SIMSv2 is a large-scale Chinese benchmark with fine-grained sentiment annotations.

We report standard MSA metrics: Acc-k (k-class sentiment classification accuracy, for $k \in 2, 3, 5, 7$), mean absolute error (MAE), and the Pearson correlation (Corr) between predicted and ground-truth sentiment scores, reporting each metric where applicable to enable fair comparison with prior work. Results across five random seeds are provided in the supplementary material.

4.2 Performance comparison

We compare STREAM with strong baselines on CMU-MOSI and CMU-MOSEI in Table 1. STREAM achieves the best reported result on Acc-7, Acc-5, Acc-2, and MAE on CMU-MOSEI, and ties the best reported Corr. In particular, the improvement in MAE suggests that the proposed routing and alignment design is effective for continuous sentiment regression in large-scale noisy settings.

Method	Acc-5↑	Acc-3↑	Acc-2↑	MAE↓	Corr↑
TFN Zadeh et al. (2017)	53.30	70.90	78.10	0.322	0.662
LMF Liu et al. (2018b)	51.60	70.00	77.80	0.327	0.651
MFN Zadeh et al. (2018a)	55.40	72.70	79.40	0.301	0.712
Graph-MFN Zadeh et al. (2018b)	48.90	68.60	76.60	0.334	0.644
MulT Tsai et al. (2019b)	54.60	74.20	80.80	0.300	0.738
MMIM Han et al. (2021)	50.50	70.40	77.80	0.339	0.641
Self-MM Yu et al. (2021)	53.50	72.70	78.70	0.315	0.691
MISA Hazarika et al. (2020)	47.50	68.90	78.20	0.342	0.671
KuDA Feng et al. (2024)	53.10	74.30	80.20	0.289	0.741
KAN-MCP Luo et al. (2025)	<u>57.30</u>	<u>75.00</u>	<u>81.60</u>	<u>0.281</u>	<u>0.742</u>
STREAM	**58.24**	**76.12**	**82.42**	**0.271**	**0.756**

Table 2: Comparison of STREAM with state-of-the-art baselines on the CH-SIMSv2 dataset. The best results in each column are highlighted in **bold**, the second-best results are <u>underlined</u>.

Configuration	Acc-7↑	Acc-5↑	Acc-2↑	MAE↓	Corr↑
Macro architecture					
w/o Hierarchical Split	54.37 ± 0.24	$55.17 \pm 0.29^{***}$	$83.08 \pm 0.18^{***}$	0.532 ± 0.0028	0.767 ± 0.0041
w/o DCCA Alignment	$53.44 \pm 0.27^{***}$	$54.07 \pm 0.31^{***}$	$80.24 \pm 0.22^{***}$	0.533 ± 0.0031	0.768 ± 0.0045
w/o Collaborative Bi-Mamba	$53.40 \pm 0.26^{***}$	$53.74 \pm 0.30^{***}$	$82.38 \pm 0.19^{***}$	$0.536 \pm 0.0034^{*}$	0.767 ± 0.0048
APMoE design					
w/o PID-inspired Experts & Loss	$53.21 \pm 0.30^{***}$	$53.86 \pm 0.34^{***}$	$82.93 \pm 0.21^{***}$	$0.541 \pm 0.0038^{***}$	$0.763 \pm 0.0052^{**}$
Average Gating Weights	54.41 ± 0.23	$55.22 \pm 0.28^{***}$	$82.33 \pm 0.20^{***}$	$0.539 \pm 0.0035^{**}$	0.769 ± 0.0047
Shared MoE Weights	$52.95 \pm 0.32^{***}$	$53.39 \pm 0.35^{***}$	$82.71 \pm 0.22^{***}$	$0.545 \pm 0.0040^{***}$	$0.759 \pm 0.0056^{**}$
Shared Router Weights	$53.59 \pm 0.28^{***}$	$54.31 \pm 0.32^{***}$	$81.24 \pm 0.23^{***}$	0.532 ± 0.0032	0.768 ± 0.0044
Full STREAM	$\mathbf{54.58 \pm 0.21}$	$\mathbf{56.10 \pm 0.25}$	$\mathbf{85.74 \pm 0.13}$	$\mathbf{0.531 \pm 0.0020}$	$\mathbf{0.772 \pm 0.0030}$

Table 3: Detailed ablation study of STREAM on CMU-MOSEI. The best-performing configuration is highlighted in **bold**. Stars indicate the significance level of each ablated variant compared with the full STREAM model ($*p < 0.05$, $**p < 0.01$, $***p < 0.001$). Results are reported as mean $\pm$ standard deviation over five independent runs.

On CMU-MOSI, STREAM remains competitive across all metrics and achieves the best Acc-5. These results suggest that combining hierarchical routing, cross-level alignment, and Mamba-based temporal fusion improves both coarse-grained and fine-grained sentiment prediction. Table 2 further shows that STREAM performs strongly on CH-SIMSv2, outperforming the compared baselines on all reported metrics. The consistent improvements across two English benchmarks and one Chinese benchmark indicate that STREAM is not restricted to a particular language or dataset scale, but provides a general multimodal fusion strategy under heterogeneous sentiment distributions. We note that these comparisons are subject to the caveat that baseline numbers are quoted rather than reproduced under a single shared feature and split protocol (Table 1).

4.3 Ablation Study

We first conduct a detailed ablation study on CMU-MOSEI, as summarized in Table 3. The results show that STREAM benefits from both its hierarchical macro-architecture and the adaptive routing design of APMoE. Removing the hierarchical split, DCCA alignment, or Collaborative Bi-Mamba consistently degrades performance, indicating that semantic-level decomposition, cross-level alignment, and temporal collaboration play complementary roles. In particular, removing DCCA reduces Acc-2 from 85.74 to 80.24, suggesting that explicitly correlating the low- and high-level representations is important for preventing semantic divergence between the two branches. Removing Collaborative Bi-Mamba also causes consistent deterioration across all five metrics, confirming that simply obtaining hierarchical representations is insufficient without an effective mechanism for consolidating their temporal information.

The APMoE-specific variants further indicate that performance gains do not arise merely from increasing model capacity. Replacing PID-inspired experts with conventional experts decreases Acc-5 from 56.10 to 53.86 and increases MAE from 0.531 to 0.541. Similarly, replacing sample-dependent gates with average gating weights degrades performance, while sharing MoE or router parameters

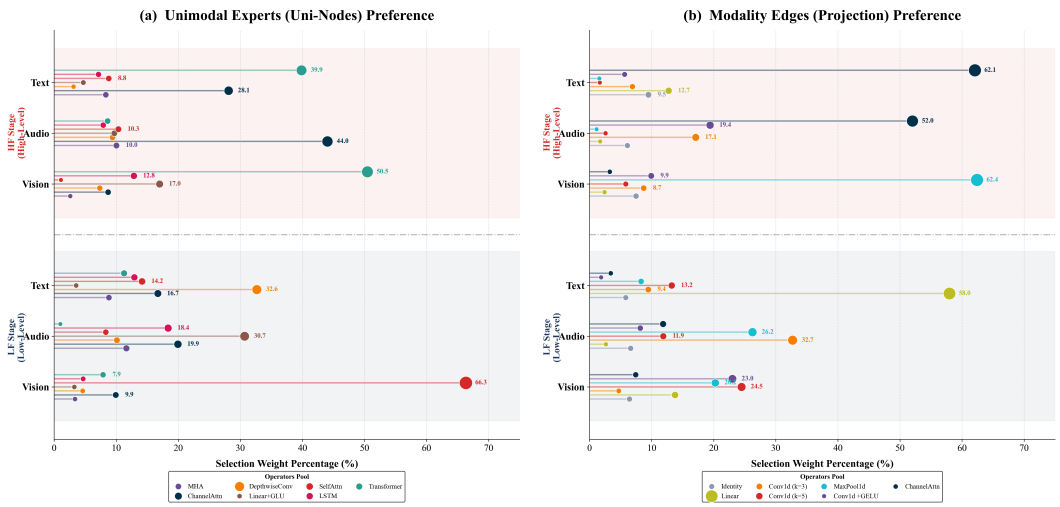


Figure 3: Visualization of the average operator weights on CMU-MOSEI. Different modalities receive different average routing mass across the candidate pool. Because routing is input-conditional (Sec. 2.3), these are dataset-level averages of per-sample decisions rather than a single discovered architecture.

across semantic levels leads to larger reductions in the fine-grained accuracy metrics Acc-7 and Acc-5. These results indicate that the low- and high-level branches exhibit different routing preferences and benefit from level-specific parameters rather than a shared or static fusion policy.

To further examine whether the design principles of STREAM generalize beyond CMU-MOSEI, we conduct additional core component ablations on CMU-MOSI and CH-SIMSv2. The detailed quantitative results and analysis are provided in the supplementary material. Across both datasets, removing individual components consistently degrades performance, demonstrating the robustness of the hierarchical decomposition, PID-inspired expert specialization, cross-level alignment, and collaborative temporal modeling.

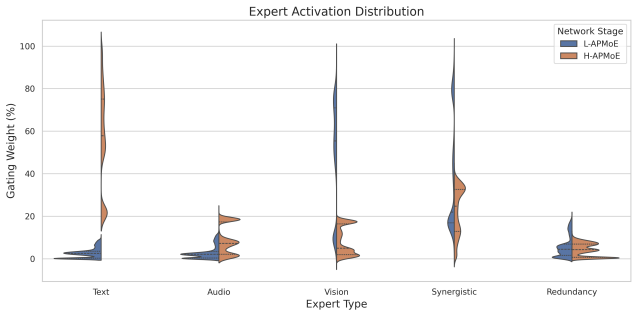


Figure 4: Violin plot illustrating the sample-level distribution of expert gating weights in L-APMoE and H-APMoE on CMU-MOSEI. The wide vertical span indicates sample-specific rather than static routing.

Taken together, the three-dataset ablations provide evidence that the major components of STREAM are complementary rather than dataset-specific. The hierarchical split establishes representations at distinct semantic granularities; the PID-inspired APMoE encourages functionally differentiated multimodal interactions; DCCA prevents the two semantic streams from becoming disconnected; and Collaborative Bi-Mamba integrates their temporal information through adaptive global–private collaboration. Importantly, no single ablation improves any reported metric over the full model, indicating that the overall architecture benefits from the joint interaction of these components rather than from an isolated module.

4.4 Interpretability and Visualization Analysis

Figure 3 shows the average operator preferences of the router. Instead of using a uniform topology, the model assigns more routing mass to sequence-modeling operators (e.g., LSTM and self-attention) for text, while favoring operators with local receptive fields (e.g., 1D convolution) for audio and vision. This difference is consistent with the heterogeneous temporal characteristics of the three modalities: textual features generally require modeling longer-range semantic dependen-

cies, whereas acoustic and visual signals contain more localized variations that can be captured by operators with stronger local inductive biases. Importantly, these preferences are not imposed manually but emerge from the input-conditional routing mechanism. As noted in Sec. 2.3, the values shown in Figure 3 are dataset-level averages over per-sample routing decisions and therefore should not be interpreted as a single fixed or discovered architecture. Rather, they summarize the aggregate computational preferences produced by the router across different samples.

While Figure 3 summarizes operator-level behavior at the dataset level, Figure 4 further reveals the sample-level dynamics of expert routing. The broad activation distributions indicate substantial variation across samples rather than nearly constant expert weights, providing evidence that the expert gate responds to heterogeneous multimodal inputs. L-APMoE shows particularly strong sample-dependent activations for the Vision and Synergy Experts, which is consistent with the role of the low-level branch in capturing localized visual cues and transient cross-modal interactions. In contrast, H-APMoE places substantially greater routing mass on the Text Expert, suggesting that higher-level representations rely more strongly on abstract linguistic semantics. At the same time, the remaining experts retain non-zero and variable activations, indicating that the high-level branch is not reduced to a purely text-only pathway. Taken together, these level-specific routing distributions are consistent with the hierarchical design of STREAM: the two semantic levels learn different expert preferences while maintaining sample-dependent variation. They also indicate that the routing mechanism does not exhibit complete expert collapse, although the observed expert weights should be interpreted as correlational routing behavior rather than causal evidence of modality importance.

To further examine the representation structure learned by STREAM, we visualize the feature distributions using t-SNE in Figure 5. Compared with the raw features, the representations produced by STREAM exhibit clearer sentiment-related organization. The low-level and high-level aligned features further show distinct but complementary distribution patterns, suggesting that the hierarchical branches capture different semantic structures while preserving sentiment-discriminative information.

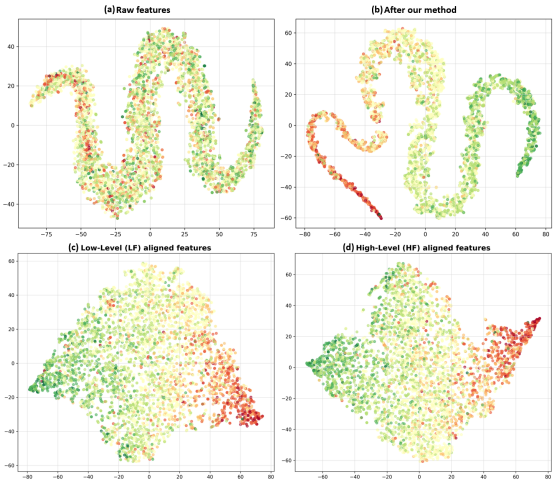


Figure 5: t-SNE visualization of feature representations on CMU-MOSEI. From (a) to (d): raw features, representations after STREAM, low-level aligned features, and high-level aligned features. Colors indicate sentiment scores.

5 Conclusion

We proposed STREAM, a hierarchical multimodal fusion framework that integrates input-conditional operator routing, PID-inspired expert specialization, cross-level alignment, and Collaborative Bi-Mamba fusion. By explicitly modeling low- and high-level representations, STREAM captures distinct cross-modal interaction patterns across semantic granularities while exposing its routing behavior through operator and expert preferences. Experiments on CMU-MOSI, CMU-MOSEI, and CH-SIMSv2 demonstrate strong performance, and cross-dataset ablations consistently validate the contributions of hierarchical decomposition, PID-inspired specialization, DCCA alignment, and collaborative temporal fusion.

STREAM may support affect-aware applications such as human–computer interaction, opinion analysis, and intelligent customer service, but several limitations remain. The perturbation-based PID objective introduces additional training cost and serves as a practical surrogate rather than an exact information-theoretic decomposition; the exposed routing weights provide correlational rather than causal interpretations; and the model relies on pre-extracted unimodal features. In addition, our evaluation is limited to English and Mandarin conversational datasets, and the two-level semantic hierarchy is manually specified. Future work will explore more efficient training, end-to-end feature learning, broader linguistic and domain settings, stronger interpretability analysis, and adaptive discovery of semantic granularity.

AI USE STATEMENT

In this work, we did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, develop or write proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement the proposed methods, translate research content, clean or reformat datasets, conduct qualitative or thematic data analysis, or interpret experimental results.

We used generative AI tools only for language editing, including improving clarity and readability, proofreading, and refining the presentation of technical descriptions. We also used generative AI tools to discuss the organization and structure of the manuscript and to improve its overall presentation. Generative AI tools were not used to generate experimental results or scientific conclusions, nor were they used to write or modify the experimental code.

All AI-assisted edits were reviewed by the authors and checked against the authors' intended technical meaning and the underlying experimental results. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available benchmark datasets (CMU-MOSI, CMU-MOSEI, and CH-SIMSv2) under their respective licenses and intended research settings. No new human-subject data were collected, and all experiments report only aggregate results without attempting to identify individual participants. We caution against the potential misuse of multimodal sentiment analysis for surveillance, profiling, or high-stakes decision-making, and intend STREAM and its associated resources solely for research purposes.

REPRODUCIBILITY STATEMENT

We ensure reproducibility of STREAM through transparent descriptions of datasets, feature extraction procedures, model configurations, and training settings. All experiments are conducted on publicly available multimodal sentiment analysis benchmarks, and the necessary implementation details are provided in the main paper and supplementary material.

1. Datasets. Experiments are conducted on three public benchmarks: CMU-MOSI, CMU-MOSEI, and CH-SIMSv2. We follow the official dataset splits and evaluation protocols for all datasets. Dataset statistics and descriptions are provided in the supplementary material.

2. Feature Extraction. Following common practice in multimodal sentiment analysis, STREAM uses pre-extracted textual, acoustic, and visual features instead of training modality encoders from raw inputs. The feature extraction pipelines and input formats are described in the supplementary material.

3. Model and Training. STREAM adopts the same architecture across all datasets, including hierarchical representation extraction, dual-level APMoE routing, DCCA-based alignment, and Collaborative Bi-Mamba fusion. The model is trained end-to-end with the joint objective consisting of regression loss, PID-inspired decoupling loss, and DCCA alignment loss. Training details and hyperparameter settings are provided in the supplementary material.

4. Evaluation. We report standard multimodal sentiment analysis metrics, including Acc-k, MAE, and Pearson correlation (Corr). All experiments are evaluated under the standard dataset protocols, and multi-run results are reported where applicable.

5. Ablation and Analysis. We provide comprehensive ablation studies and additional analyses to evaluate the contributions of hierarchical representation learning, PID-inspired expert routing, cross-level alignment, and Collaborative Bi-Mamba fusion. Further analysis includes routing visualization, expert behavior analysis, and robustness evaluation.

6. Code Release. We provide the implementation details, training configurations, and evaluation scripts required to reproduce the reported results. The complete code and associated resources will be released upon acceptance.

REFERENCES

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2): 423–443, 2019.

Yiyang Fang, Wenke Huang, Guancheng Wan, Kehua Su, and Mang Ye. Emoe: Modality-specific enhanced dynamic emotion experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14314–14324, 2025.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. In *Journal of Machine Learning Research*, volume 23, pp. 1–39, 2022.

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 Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14755–14766, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.865.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Tianming Han, Zhijie Pan, and Wenchi Ge. Multimodal multi-granularity fusion model with mamba architecture for ames mutagenicity prediction. *Journal of Medicinal Chemistry*, 2026.

Wei Han, Hui Chen, and Soujanya Poria. Improving multimodal fusion with hierarchical mutual information maximization for multimodal sentiment analysis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9180–9192. Association for Computational Linguistics, 2021.

Devamanyu Hazarika, Roger Zimmermann, and Soujanya Poria. Misa: Modality-invariant and -specific representations for multimodal sentiment analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1122–1131, 2020.

Xilin He, Haijian Liang, Boyi Peng, Weicheng Xie, Muhammad Haris Khan, Siyang Song, and Zitong Yu. Msamba: Exploring multimodal sentiment analysis with state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Shenjie Jiang, Zhuoyu Wang, Xuecheng Wu, Hongru Ji, Mingxin Li, Xianghua Li, and Chao Gao. Ddse: A decoupled dual-stream enhanced framework for multimodal sentiment analysis with text-centric ssm. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 5893–5902, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3754817.

X. Li et al. Hybrid cnn-mamba enhancement network for robust multimodal sentiment analysis. *arXiv preprint arXiv:2507.23444*, 2025.

Yan Li, Yifei Xing, Xiangyuan Lan, Xin Li, Haifeng Chen, and Dongmei Jiang. Alignmamba-2: Enhancing multimodal fusion and sentiment analysis with modality-aware mamba. *arXiv preprint arXiv:2603.18462*, 2026.

Yong Li, Yuanzhi Wang, and Zhen Cui. Decoupled multimodal distilling for emotion recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6631–6640, June 2023.

Paul Pu Liang, Yun Cheng, Xiang Fan, Chenhao Ling, Shen Nie, Richard Chen, Zhiying Deng, Nicholas Allen, Randy Auerbach, Faisal Mahmood, et al. Quantifying & modeling multimodal interactions: An information decomposition framework. *Advances in Neural Information Processing Systems*, 36, 2023.

Paul Pu Liang, Zhiying Deng, Mike Q Ma, James Y Zou, Louis-Philippe Morency, and Ruslan Salakhutdinov. Factorized contrastive learning: Going beyond multi-view redundancy. *Advances in Neural Information Processing Systems*, 36, 2024.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2018a.

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. 2022.

Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminalshmanan, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Efficient low-rank multimodal fusion with modality-specific factors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2247–2256. Association for Computational Linguistics, 2018b.

Miaosen Luo, Yuncheng Jiang, and Sijie Mai. Towards explainable fusion and balanced learning in multimodal sentiment analysis. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 1997–2006, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3755591.

Md Mahfuzur Rahman et al. Mag-bert: Text representation learning via masked attention and gated multimodal fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020a.

Md Mahfuzur Rahman et al. Integrating multimodal information in large pre-trained transformers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020b.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

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 *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6558–6569, 2019a.

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, 2019b.

Paul L Williams and Randall D Beer. Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*, 2010.

Patricia Wollstadt, Sebastian Schmitt, and Michael Wibral. A rigorous information-theoretic definition of redundancy and relevancy in feature selection based on (partial) information decomposition. *Journal of Machine Learning Research*, 24(131):1–44, 2023.

Han Wu, Yanming Sun, Yunhe Yang, and Derek F Wong. Beyond simple fusion: Adaptive gated fusion for robust multimodal sentiment analysis. *arXiv preprint arXiv:2510.01677*, 2025.

Wenmeng Yu, Hua Xu, Ziqi Yuan, and Jiele Wu. Learning modality-specific representations with self-supervised multi-task learning for multimodal sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10790–10797, 2021.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in videos. *arXiv preprint arXiv:1606.06259*, 2016.

Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1103–1114. Association for Computational Linguistics, 2017.

Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Memory fusion network for multi-view sequential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018a.

Amir Zadeh, Paul Pu Liang, Soujanya Poria, E. Cambria, and Louis philippe Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Annual Meeting of the Association for Computational Linguistics*, 2018b.

AmirAli Bagher 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 *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2236–2246, 2018c.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

Appendix

Throughout this supplementary material, APMoE denotes Automated Partial-Information Mixture-of-Experts, PID denotes Partial Information Decomposition, DCCA denotes Deep Canonical Correlation Analysis, and NAS denotes Neural Architecture Search. In the terminology of the main paper, the learned mechanism is input-conditional operator routing over a differentiable search space rather than conventional architecture search that produces a single fixed topology.

For consistency with the notation in the main paper, we use $\mathbf{X}_m^{\mathrm{LL}}$ and $\mathbf{X}_m^{\mathrm{HL}}$ to denote the low-level and high-level representations of modality $m \in \{t, a, v\}$ before APMoE fusion. The outputs of L-APMoE and H-APMoE are denoted by $\mathbf{X}_{\mathrm{LL}}$ and $\mathbf{X}_{\mathrm{HL}}$, respectively. The outputs of the Uniqueness, Synergy, and Redundancy Experts used by the PID-inspired auxiliary objective are denoted by $\mathbf{Y}^u$, $\mathbf{Y}^s$, and $\mathbf{Y}^r$, consistent with Sec. 3.4 of the main paper. DCCA projects $\mathbf{X}_{\mathrm{LL}}$ and $\mathbf{X}_{\mathrm{HL}}$ into the aligned latent representations $\mathbf{Z}_{\mathrm{L}}$ and $\mathbf{Z}_{\mathrm{H}}$, which are subsequently provided to Collaborative Bi-Mamba. No independent classification head is used; all classification metrics are obtained by discretizing the continuous regression prediction.

A Cross-dataset Core Component Ablation

To evaluate the generalization of STREAM beyond CMU-MOSEI, we perform core component ablations on CMU-MOSI and CH-SIMSv2. The results consistently demonstrate that each module contributes complementary benefits. On CMU-MOSI, removing the hierarchical split decreases Acc-2 from 86.32 to 84.17, while removing the PID-inspired loss and experts reduces Acc-5 from 56.15 to 53.82 and Corr from 0.790 to 0.772. Removing DCCA produces the largest degradation in Acc-2, from 86.32 to 81.54, indicating the importance of cross-level alignment. Collaborative Bi-Mamba is also beneficial, with its removal decreasing Acc-5 to 53.61 and increasing MAE from 0.711 to 0.726.

A similar pattern is observed on CH-SIMSv2. Removing the hierarchical split reduces Acc-2 from 82.42 to 80.39, whereas removing the PID-inspired loss and experts reduces Acc-5 from 58.24 to 56.55 and increases MAE from 0.271 to 0.284. The DCCA ablation decreases Acc-2 from 82.42 to 77.91, further supporting the importance of coordinating low- and high-level semantic representations. Removing Collaborative Bi-Mamba also degrades both classification and regression performance.

B Model Complexity and Inference Performance Benchmark

Table 6 presents the computational complexity, storage overhead, and inference efficiency of STREAM. Despite its dynamic routing design, STREAM remains relatively compact, with only 1.77M parameters and a 6.76 MB model size. All measurements are conducted on an NVIDIA RTX 4070Ti graphics processing unit (GPU) with Python 3.10, PyTorch 2.0.1, and Compute Unified Device Architecture (CUDA) 11.7.

Configuration	Acc-7↑	Acc-5↑	Acc-2↑	MAE↓	Corr↑
Full STREAM	**47.91 ± 0.21**	**56.15 ± 0.25**	**86.32 ± 0.13**	**0.711 ± 0.0021**	**0.790 ± 0.0032**
w/o Hierarchical Split	$47.48 \pm 0.24^{*}$	$55.09 \pm 0.29^{***}$	$84.17 \pm 0.19^{***}$	$0.716 \pm 0.0028^{*}$	$0.783 \pm 0.0041^{*}$
w/o PID-inspired Experts & Loss	$46.35 \pm 0.31^{***}$	$53.82 \pm 0.35^{***}$	$83.41 \pm 0.22^{***}$	$0.729 \pm 0.0038^{***}$	$0.772 \pm 0.0054^{***}$
w/o DCCA	$46.51 \pm 0.28^{***}$	$53.96 \pm 0.33^{***}$	$81.54 \pm 0.24^{***}$	$0.721 \pm 0.0034^{***}$	$0.778 \pm 0.0049^{**}$
w/o Collaborative Bi-Mamba	$46.42 \pm 0.30^{***}$	$53.61 \pm 0.36^{***}$	$83.06 \pm 0.23^{***}$	$0.726 \pm 0.0037^{***}$	$0.775 \pm 0.0051^{***}$

Table 4: Core component ablations of STREAM on CMU-MOSI. The best-performing configuration is highlighted in **bold**. Removing any major component consistently degrades performance, demonstrating that hierarchical decomposition, PID-inspired expert specialization, cross-level alignment, and collaborative temporal modeling contribute complementary benefits. Stars indicate the significance level of each ablated variant compared with the full STREAM model ($*p < 0.05$, $**p < 0.01$, $***p < 0.001$). Results are reported as mean ± standard deviation over five independent runs.

Configuration	Acc-5↑	Acc-3↑	Acc-2↑	MAE↓	Corr↑
Full STREAM	**58.24 ± 0.24**	**76.12 ± 0.22**	**82.42 ± 0.15**	**0.271 ± 0.0022**	**0.756 ± 0.0035**
w/o Hierarchical Split	$57.76 \pm 0.27^{*}$	$75.04 \pm 0.26^{***}$	$80.39 \pm 0.19^{***}$	$0.275 \pm 0.0029^{*}$	$0.750 \pm 0.0043^{*}$
w/o PID-inspired Experts & Loss	$56.55 \pm 0.32^{***}$	$73.81 \pm 0.31^{***}$	$79.62 \pm 0.23^{***}$	$0.284 \pm 0.0040^{***}$	$0.741 \pm 0.0057^{**}$
w/o DCCA	$56.82 \pm 0.30^{***}$	$73.69 \pm 0.33^{***}$	$77.91 \pm 0.25^{***}$	$0.278 \pm 0.0036^{**}$	$0.746 \pm 0.0050^{**}$
w/o Collaborative Bi-Mamba	$56.73 \pm 0.31^{***}$	$73.47 \pm 0.34^{***}$	$79.28 \pm 0.24^{***}$	$0.281 \pm 0.0038^{**}$	$0.743 \pm 0.0053^{**}$

Table 5: Core component ablations of STREAM on CH-SIMSv2. The best-performing configuration is highlighted in **bold**. The consistent degradation after removing individual components demonstrates that the effectiveness of STREAM generalizes across different languages and sentiment annotation settings. Stars indicate the significance level of each ablated variant compared with the full STREAM model ($*p < 0.05$, $**p < 0.01$, $***p < 0.001$). Results are reported as mean ± standard deviation over five independent runs.

C Algorithm

Algorithm 1 Training Procedure of STREAM

Require: Training set $\mathcal{D}$; model parameters Θ; loss weights $\lambda_{\text{pid}}, \lambda_{\text{dcca}}$

while not converged **do**
  for each mini-batch (X^t, X^a, X^v, y) in $\mathcal{D}$ **do**
    // Input projection and hierarchical stream construction
    $(H^t, H^a, H^v) \leftarrow \text{Project}(X^t, X^a, X^v)$
    for each modality $m \in \{t, a, v\}$ **do**
      $(X_m^{\text{LL}}, X_m^{\text{HL}}) \leftarrow \text{HierarchicalExtractor}(H^m)$
    end for
    // Low-level APMoE
    $(X_{\text{LL}}, Y_{\text{full}}^{u,\text{LL}}, Y_{\text{full}}^{s,\text{LL}}, Y_{\text{full}}^{r,\text{LL}}) \leftarrow \text{L-APMoE}(\{X_m^{\text{LL}}\}_{m\in\{t,a,v\}})$
    $\mathcal{L}_{\text{pid}}^{\text{LL}} \leftarrow \text{PIDLoss}(Y_{\text{full}}^{u,\text{LL}}, Y_{\text{full}}^{s,\text{LL}}, Y_{\text{full}}^{r,\text{LL}})$
    // High-level APMoE
    $(X_{\text{HL}}, Y_{\text{full}}^{u,\text{HL}}, Y_{\text{full}}^{s,\text{HL}}, Y_{\text{full}}^{r,\text{HL}}) \leftarrow \text{H-APMoE}(\{X_m^{\text{HL}}\}_{m\in\{t,a,v\}})$
    $\mathcal{L}_{\text{pid}}^{\text{HL}} \leftarrow \text{PIDLoss}(Y_{\text{full}}^{u,\text{HL}}, Y_{\text{full}}^{s,\text{HL}}, Y_{\text{full}}^{r,\text{HL}})$
    // Cross-level DCCA alignment
    $(Z_{\text{L}}, Z_{\text{H}}, \mathcal{L}_{\text{dcca}}) \leftarrow \text{DCCA}(X_{\text{LL}}, X_{\text{HL}})$
    // Collaborative temporal fusion
    $(X_{\text{LL}}^{\text{out}}, X_{\text{HL}}^{\text{out}}) \leftarrow \text{CollaborativeBiMamba}(Z_{\text{L}}, Z_{\text{H}})$
    $r \leftarrow \text{Pool}(X_{\text{LL}}^{\text{out}}, X_{\text{HL}}^{\text{out}})$
    $\hat{y} \leftarrow \text{Regressor}(r)$
    // Joint optimization
    $\mathcal{L}_{\text{reg}} \leftarrow \text{HuberLoss}(\hat{y}, y)$
    $\mathcal{L}_{\text{pid}} \leftarrow \mathcal{L}_{\text{pid}}^{\text{LL}} + \mathcal{L}_{\text{pid}}^{\text{HL}}$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{reg}} + \lambda_{\text{pid}}\mathcal{L}_{\text{pid}} + \lambda_{\text{dcca}}\mathcal{L}_{\text{dcca}}$
    Update Θ by backpropagation on $\mathcal{L}$
  end for
end while

Metric	Value
Total Parameters	1.77 M
Trainable Parameters	1.77 M
Model Size on Disk	6.76 MB
Single-Batch Inference Latency	120.74 ms
Throughput	457.38 Samples / Second

Table 6: Model Complexity and Inference Performance Benchmark.

Algorithm 1 follows exactly the computational order used in the main paper. The input textual, acoustic, and visual sequences are first projected and converted into modality-specific low-level and high-level representations $\mathbf{X}_m^{\mathrm{LL}}$ and $\mathbf{X}_m^{\mathrm{HL}}$. L-APMoE and H-APMoE then produce the fused semantic-level representations $\mathbf{X}_{\mathrm{LL}}$ and $\mathbf{X}_{\mathrm{HL}}$, respectively.

In parallel, the Uniqueness, Synergy, and Redundancy Experts produce $\mathbf{Y}^u$, $\mathbf{Y}^s$, and $\mathbf{Y}^r$ under both the full-modality input and the modality-perturbed inputs. These expert representations are used only to construct the PID-inspired auxiliary loss defined in Sec. 3.4 of the main paper.

The two APMoE outputs $\mathbf{X}_{\mathrm{LL}}$ and $\mathbf{X}_{\mathrm{HL}}$ are subsequently aligned by DCCA, which produces $\mathbf{Z}_{\mathrm{L}}$ and $\mathbf{Z}_{\mathrm{H}}$. These aligned representations are then processed by Collaborative Bi-Mamba. After temporal aggregation, a regression head produces the continuous sentiment prediction $\hat{y}$. STREAM does not employ an independent classification head; classification metrics are computed only by discretizing $\hat{y}$ during evaluation.

D Multi-seed Stability Analysis

Run	**Acc-7↑**	**Acc-5↑**	**Acc-2↑**	**MAE↓**	**Corr↑**
1	54.300	56.312	85.637	0.5334	0.7686
2	54.477	55.817	85.921	0.5296	0.7739
3	54.561	56.147	85.598	0.5318	0.7761
4	54.720	56.359	85.805	0.5284	0.7714
5	54.843	55.864	85.740	0.5318	0.7701
Mean ± Std.	**54.58 ± 0.21**	**56.10 ± 0.25**	**85.74 ± 0.13**	**0.5310 ± 0.0020**	**0.7720 ± 0.0030**

Table 7: Performance variation of STREAM across five independent runs on CMU-MOSEI. The reported run indices represent different random initialization settings; the exact random seeds are determined by the implementation configuration. Mean and standard deviation are computed over the five runs.

To evaluate the robustness of STREAM against random initialization and training stochasticity, we conduct five independent runs on the CMU-MOSEI dataset under the same experimental configuration. The run indices in Table 7 denote different random initialization settings, while the exact numerical seed values are determined by the implementation configuration.

As shown in Table 7, STREAM exhibits stable performance across different runs. The standard deviations of all reported metrics are small, with only 0.21, 0.25, and 0.13 percentage points for Acc-7, Acc-5, and Acc-2, respectively. The regression metrics are also consistent, with a standard deviation of 0.0020 for MAE and 0.0030 for Corr. These results indicate that the observed performance is stable across the five reported runs.

E Additional Analysis on CMU-MOSEI

E.1 Robustness to missing modalities

Table 8 evaluates test-time behavior under incomplete multimodal observations. Specifically, we randomly replace the features of a target modality with zeros for a given proportion of test samples while keeping all model parameters unchanged. This setting measures the sensitivity of the learned multimodal representation to incomplete observations without additional retraining.

Missing rate	Acc-7 ↑	Acc-5 ↑	Acc-2 ↑	MAE ↓	Corr ↑
0% missing	**54.58 ± 0.21**	**56.10 ± 0.25**	**85.74 ± 0.13**	**0.531 ± 0.0020**	**0.772 ± 0.0030**
Text modality					
10%	$52.14 \pm 0.28^{***}$	$53.69 \pm 0.31^{***}$	$81.63 \pm 0.21^{***}$	$0.588 \pm 0.0032^{***}$	$0.709 \pm 0.0048^{***}$
15%	$51.11 \pm 0.30^{***}$	$52.53 \pm 0.34^{***}$	$80.02 \pm 0.24^{***}$	$0.615 \pm 0.0037^{***}$	$0.672 \pm 0.0054^{***}$
30%	$47.22 \pm 0.36^{***}$	$48.33 \pm 0.39^{***}$	$75.41 \pm 0.29^{***}$	$0.696 \pm 0.0045^{***}$	$0.572 \pm 0.0062^{***}$
50%	$40.44 \pm 0.43^{***}$	$41.24 \pm 0.46^{***}$	$66.33 \pm 0.35^{***}$	$0.825 \pm 0.0052^{***}$	$0.434 \pm 0.0071^{***}$
Vision modality					
10%	54.48 ± 0.22	55.84 ± 0.27	$85.09 \pm 0.15^{***}$	0.534 ± 0.0023	0.770 ± 0.0034
15%	$54.18 \pm 0.23^{*}$	$55.22 \pm 0.28^{***}$	$84.48 \pm 0.17^{***}$	$0.535 \pm 0.0025^{*}$	$0.766 \pm 0.0037^{*}$
30%	$53.75 \pm 0.26^{***}$	$54.60 \pm 0.30^{***}$	$84.00 \pm 0.19^{***}$	$0.539 \pm 0.0029^{***}$	$0.758 \pm 0.0042^{***}$
50%	$53.39 \pm 0.29^{***}$	$53.31 \pm 0.33^{***}$	$82.62 \pm 0.22^{***}$	$0.547 \pm 0.0033^{***}$	$0.751 \pm 0.0046^{***}$
Audio modality					
10%	54.45 ± 0.21	55.93 ± 0.26	85.59 ± 0.14	0.532 ± 0.0022	0.772 ± 0.0033
15%	$54.13 \pm 0.23^{*}$	$55.60 \pm 0.27^{*}$	$85.27 \pm 0.16^{***}$	0.534 ± 0.0024	0.768 ± 0.0036
30%	$53.37 \pm 0.25^{***}$	$54.99 \pm 0.29^{***}$	$84.84 \pm 0.18^{***}$	$0.541 \pm 0.0028^{***}$	$0.763 \pm 0.0040^{**}$
50%	$52.76 \pm 0.28^{***}$	$53.62 \pm 0.32^{***}$	$83.94 \pm 0.21^{***}$	$0.547 \pm 0.0032^{***}$	$0.757 \pm 0.0045^{***}$

Table 8: Performance comparison of STREAM under different modality missing rates on CMU-MOSEI. The complete-input setting is highlighted in **bold**. Stars indicate the significance level of each missing-modality setting compared with the full-input STREAM model ($*p < 0.05$, $**p < 0.01$, $***p < 0.001$). Results are reported as mean ± standard deviation over five independent runs.

Performance generally decreases as the missing rate increases, but the degradation differs substantially across modalities. Missing text produces the most pronounced decline: at a 50% missing rate, Acc-2 falls from 85.74 to 66.33, and Corr decreases from 0.772 to 0.434. In contrast, the degradation is smaller when visual or acoustic inputs are removed; with 50% missing vision or audio, Acc-2 remains 82.62 and 83.94, respectively.

E.2 Alignment Analysis

We further compare the full STREAM model with a variant that replaces the final Collaborative Bi-Mamba module with a cross-attention (Cross-Attn) fusion block. As reported in Table 9, the Bi-Mamba version produces higher cross-level cosine similarity and lower maximum mean discrepancy (MMD) after the final fusion stage. These values are reported as representation-level alignment diagnostics between the two semantic streams.

Alignment Type	Processing Stage	Bi-Mamba		Cross-Attn	
		Cosine ↑	MMD ↓	Cosine ↑	MMD ↓
Cross-Modal	Initial Input	0.0152	1.1845	0.0152	1.1845
	After L-APMoE	0.2423	0.6832	0.3846	1.4443
	After H-APMoE	0.5587	0.4210	0.2269	1.3437
Cross-Level	Before Final Fusion	0.0961	2.1890	0.0504	1.5723
	After Final Fusion	0.5654	0.6640	0.3668	0.9232

Table 9: Quantitative alignment analysis under different final fusion modules on CMU-MOSEI. We report cosine similarity and MMD distance for both cross-modal alignment and cross-level consistency.

Here, L-APMoE and H-APMoE denote the low-level and high-level APMoE branches, respectively. Consistent with the notation in the main paper, their fused outputs are $\mathbf{X}_{\mathrm{LL}}$ and $\mathbf{X}_{\mathrm{HL}}$. The "Before Final Fusion" statistics are calculated from the aligned low-level and high-level representations supplied to the final fusion module, whereas the "After Final Fusion" statistics are calculated from the corresponding outputs of the final fusion module.

We also observe noticeable differences in the intermediate cross-modal statistics. In particular, the full STREAM model attains stronger alignment statistics after H-APMoE in this diagnostic. Because the complete model is optimized end-to-end, changing the final fusion module can also alter the representations learned in the preceding stages. Therefore, these results are interpreted as

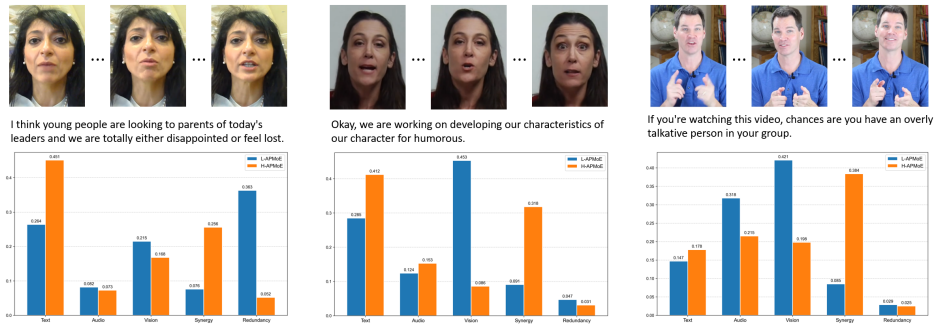


Figure 6: Sample-level case study of dynamic routing. From left to right: (a) Noise filtering and modality compensation; (b) Micro-expression capture versus global semantic smoothing; (c) Resolving semantic ambiguity in emotionally incongruent expressions.

evidence of different representation-alignment behavior rather than as a standalone measurement of predictive superiority.

E.3 Qualitative Analysis of Sample-level Routing Behaviors

To illustrate the decision process of the hierarchical architecture, we analyze three representative cases in Figure 6. In a noisy environment with flat vocal delivery (Figure 6(a)), L-APMoE assigns greater weight to the Redundancy Expert, while H-APMoE emphasizes the Text Expert and produces a negative prediction. For a neutral utterance with a transient micro-expression (Figure 6(b)), L-APMoE assigns greater weight to the Vision Expert, whereas H-APMoE emphasizes the Text and Synergy Experts, resulting in a slightly positive prediction. In an incongruent case (Figure 6(c)) with negative text but enthusiastic delivery, both branches downweight the Text Expert; L-APMoE emphasizes visual and acoustic cues, while H-APMoE relies more strongly on the Synergy Expert and produces a positive prediction. These three cases are illustrative and selected for exposition; they are not evidence of a general pattern.

E.4 Routing Visualization

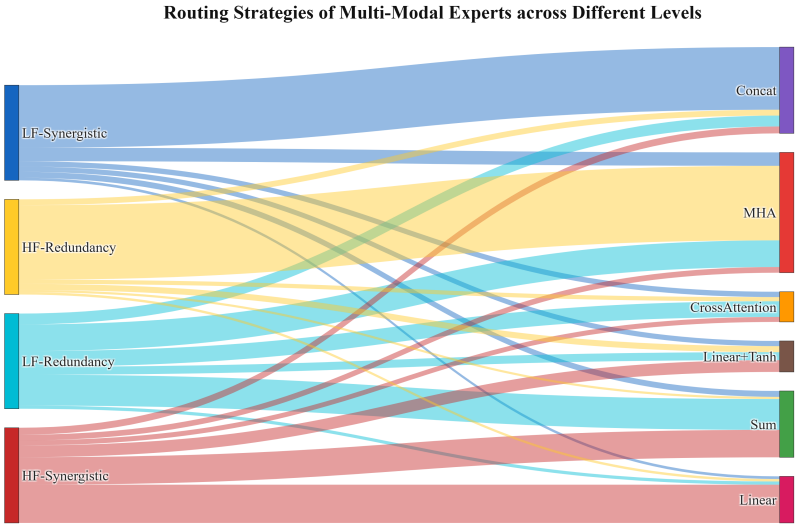


Figure 7: Sankey visualization of the routing preferences of hierarchical multimodal experts on CMU-MOSEI. The diagram traces how the Synergy Expert and Redundancy Expert at different semantic levels are routed to candidate fusion operators.

To further reveal the internal routing behavior of STREAM, we visualize the expert-to-operator assignment patterns using a Sankey diagram in Figure 7. Specifically, we focus on the two multimodal interaction experts, namely the Synergy Expert and the Redundancy Expert, in both the low-level and high-level branches.

The visualization shows that these experts do not collapse to a single shared routing pattern across semantic levels. Instead, the low-level and high-level branches exhibit different operator preferences, consistent with the use of separate level-specific routing parameters in L-APMoE and H-APMoE. The routing distributions of the Synergy Expert and Redundancy Expert also differ, indicating behavioral differentiation between the two PID-inspired expert types.

E.5 Expert Behavior Analysis

We additionally provide a quantitative diagnostic to examine whether the learned PID-inspired experts exhibit distinct functional behaviors beyond qualitative routing visualizations.

To probe the intended expert roles, we analyze three complementary properties: representation divergence, perturbation sensitivity under modality masking, and consistency across masked views. The results are summarized in Tables 10 and 11. In these tables, Uni, Syn, and Red denote Uniqueness, Synergy, and Redundancy representations, respectively; Cos denotes cosine similarity; and "mask" denotes evaluation under modality-masked views. The symbols ΔSyn and ΔRed denote the response shifts of the Synergy and Redundancy Experts under modality masking.

Branch	**Uni-Syn Cos**	**Uni-Red Cos**	**Syn-Red Cos**	**Uni-Syn MMD**	**Uni-Red MMD**	**Syn-Red MMD**
Low-level	-0.7034	0.2269	-0.2427	2.7547	2.3297	3.1015
High-level	-0.9148	0.3952	-0.5011	3.9466	3.0249	3.6018

Table 10: Expert representation diagnostics on CMU-MOSEI. We report cosine similarity and MMD between Uniqueness, Synergy, and Redundancy expert representations for the low-level and high-level branches. These quantities characterize geometric divergence between expert representations and are not interpreted as direct information-theoretic PID quantities.

Branch	**ΔSyn / ΔRed**	**mask Syn Cos / MMD**	**mask Red Cos / MMD**
Low-level	0.5236 / 0.1078	-0.1031 / 1.0684	0.8853 / 0.1891
High-level	0.6654 / 0.0543	-0.2513 / 1.3631	0.9369 / 0.1174

Table 11: Perturbation-sensitivity and masking-stability analysis on CMU-MOSEI. We report the average response shift of the Synergy and Redundancy Experts under modality masking, together with their consistency across different masked views. A larger Δ indicates a larger representation change, while higher cosine similarity and lower MMD indicate greater consistency across masked inputs.

First, the expert representations exhibit clear geometric differences. In the low-level branch, the cosine similarity between the Uniqueness and Synergy representations is -0.7034, while that between the Synergy and Redundancy representations is -0.2427. Their MMD distances are also non-trivial, with the Synergy–Redundancy pair reaching 3.1015. A similar pattern is observed in the high-level branch. These results indicate that the different expert groups do not collapse to numerically identical representations.

Second, the perturbation-sensitivity analysis shows that the Synergy Expert changes more strongly under modality corruption than the Redundancy Expert. In the low-level branch, the average response shift of the Synergy Expert is 0.5236, compared with 0.1078 for the Redundancy Expert. This behavioral difference is consistent with the intended surrogate objective: the Synergy Expert is encouraged to respond to joint multimodal evidence, whereas the Redundancy Expert is encouraged to retain information that is stable across different masked views.

Third, the cross-mask statistics show substantially greater stability for the Redundancy Expert. In the low-level branch, the Redundancy Expert obtains a cross-mask cosine similarity of 0.8853 with an MMD of 0.1891, whereas the Synergy Expert obtains a cosine similarity of -0.1031 and an MMD of 1.0684. The same behavioral contrast is observed in the high-level branch.

Taken together, the representation-space diagnostics, masking response shifts, and cross-mask consistency support behavioral differentiation between the PID-inspired expert groups. These analyses are used as empirical diagnostics of the surrogate objectives and are not claimed to constitute an exact information-theoretic PID decomposition.

F Ablation Studies

Clarification of search-stage and final configurations. During the candidate-space analysis, "full" refers to the unpruned candidate pool under the corresponding pool-size setting. These experiments are used to analyze operator redundancy and do not denote the final STREAM configuration. After operator-level pruning, the refined candidate set is used in the final STREAM model reported in the main paper.

F.1 Candidate Pool Design

We first investigate the sensitivity of STREAM to the routing graph configuration by varying the number of selected expert nodes and edge operations on CMU-MOSEI. As shown in Table 12, the overall performance remains relatively stable across different hyperparameter choices. Increasing the number of nodes or edges does not lead to monotonic improvements, suggesting that a larger active routing set is not necessarily beneficial once sufficient operator diversity is available.

In the following tables, Acc-k denotes k-class accuracy, F1 denotes the F1 score, MAE denotes mean absolute error, and Corr denotes the Pearson correlation coefficient.

Nodes	Edges	Acc-7↑	Acc-5↑	Acc-2↑	F1↑	MAE↓	Corr↑
1	1	53.74	55.24	86.33	86.33	0.539	0.766
1	2	54.27	55.84	85.60	85.60	0.540	0.766
1	3	54.58	56.24	87.06	87.06	0.543	0.762
1	4	53.80	55.17	84.12	84.12	0.544	0.763
1	5	**54.81**	**56.37**	86.83	86.83	0.539	0.766
2	1	53.38	54.93	85.95	85.95	0.532	0.768
2	2	54.69	56.31	84.77	84.76	0.539	0.767
2	3	54.60	56.15	85.60	85.60	0.533	0.768
2	4	54.76	56.31	85.04	85.04	0.540	0.765
2	5	54.29	55.77	84.71	84.71	0.542	0.759
3	1	54.33	55.79	**87.40**	**87.40**	0.536	**0.772**
3	2	54.27	55.79	87.35	87.35	0.543	0.767
3	3	53.76	54.95	87.02	87.03	0.542	0.770
3	4	54.18	55.32	86.98	86.98	0.542	0.771
3	5	54.25	55.53	87.02	87.02	0.542	0.769
4	1	54.58	56.10	85.74	85.73	**0.531**	**0.772**
4	2	54.25	55.79	86.15	86.16	0.539	0.768
4	3	54.27	55.79	85.00	85.00	0.540	0.769
4	4	54.25	55.86	85.13	85.13	0.539	0.768
4	5	54.29	55.88	84.93	84.93	0.535	0.768

Table 12: Parameter sensitivity analysis on CMU-MOSEI under different top-n node and top-m edge settings. Best values are highlighted in **bold**.

Among all configurations, we select the $(4\ \text{nodes}, 1\ \text{edge})$ configuration according to the validation-set regression objective, prioritizing MAE and Corr rather than any individual classification metric. Therefore, $(\text{Nodes} = 4, \text{Edges} = 1)$ is used as the default routing configuration in the main experiments.

Building on the chosen routing configuration, we provide a more detailed analysis of the candidate operator pools used by the input-conditional operator router. The study proceeds in two stages. We first perform a coarse-grained candidate-pool ablation to determine suitable pool sizes for the

edge operators and unimodal experts. We then conduct fine-grained leave-one-operator-out analyses within the selected pools.

In Table 13, MHA denotes multi-head attention, GLU denotes gated linear unit, LSTM denotes long short-term memory, GRU denotes gated recurrent unit, GELU denotes Gaussian error linear unit, FFN denotes feed-forward network, and MLP denotes multilayer perceptron. The abbreviations `conv`, `dwconv`, `avgpool`, `maxpool`, and `attn` denote one-dimensional convolution, depthwise one-dimensional convolution, average pooling, maximum pooling, and attention, respectively. A number following `conv`, `dwconv`, or a pooling operator denotes its kernel size, while `d2` denotes a dilation rate of two.

Pool Type	**Scale**	**Operators**
Edge Operator Pool	Small	`identity`, `linear`, `conv3`, `dwconv3`, `maxpool3`, `channel_attn`, `temporal_shift`
	Medium	`temporal_diff`, `dwconv3`, `temporal_shift`, `identity`, `conv5`, `Conv1d+GELU`, `conv3`, `linear`, `maxpool3`, `channel_attn`
	Large	`identity`, `linear`, `Conv1d+GELU`, `residual_mlp`, `conv3`, `conv5`, `dilated_conv3_d2`, `dwconv3`, `dwconv5`, `avgpool3`, `avgpool5`, `maxpool3`, `maxpool5`, `channel_attn`, `self_attn`, `temporal_shift`, `temporal_diff`
Uni-modal Expert Pool	Basic	`mha`, `dwconv3`, `linear_glu`, `lstm`, `channel_attn`, `self_attn`, `transformer`
	Medium	`identity`, `linear`, `linear_glu`, `channel_attn`, `self_attn`, `mha`, `transformer`, `conv3`, `dwconv3`, `dilated_conv3_d2`, `lstm`, `gru`, `temporal_shift`, `temporal_diff`
	Large	`identity`, `linear`, `ffn_relu`, `glu`, `linear_glu`, `channel_attn`, `self_attn`, `mha`, `transformer`, `conv3`, `conv5`, `dwconv3`, `dwconv5`, `dilated_conv3_d2`, `avgpool3`, `maxpool3`, `lstm`, `gru`, `temporal_shift`, `temporal_diff`
Multi-modal Expert Pool	Large	`self_attn`, `CrossAttention`, `linear`, `linear+tanh`, `sum`, `concat`, `MHA`

Table 13: Detailed definitions of the candidate operator pools used in the pool-size ablation. We explicitly list the operator composition of each pool.

We begin by studying how the size of the candidate pools affects performance. For the edge operators and unimodal experts, we compare small/basic, medium, and large variants. The exact compositions are summarized in Table 13, and the corresponding results are reported in Table 14.

Pool Type	**Scale**	**Acc-7↑**	**Acc-5↑**	**Acc-2↑**	**F1↑**	**MAE↓**	**Corr↑**
Edge Operator Pool	Small	52.43	54.19	83.04	83.04	0.540	0.760
	Medium	**53.04**	**55.13**	83.36	83.36	**0.539**	**0.763**
	Large	52.27	53.99	**84.11**	**84.11**	0.544	0.760
Uni-modal Expert Pool	Basic	**52.85**	**55.02**	83.65	83.65	**0.538**	0.759
	Medium	52.42	54.21	**85.46**	**85.45**	0.553	0.756
	Large	52.24	54.41	83.71	83.71	0.544	**0.761**

Table 14: Preliminary ablation on candidate pool size for edge operators and unimodal experts on CMU-MOSEI. Best values within each pool type are highlighted in **bold**.

For the edge-operator pool, the medium configuration provides the most balanced behavior. Although the large pool yields the best Acc-2/F1, the medium pool obtains the best MAE and Corr together with the strongest Acc-5 and Acc-7. We therefore use the *medium* edge-operator pool before the subsequent operator-level pruning stage.

For the unimodal expert pool, the basic configuration achieves the best MAE and the strongest Acc-5/Acc-7 while retaining competitive Corr. The medium pool improves Acc-2/F1 but exhibits worse regression metrics, and the large pool obtains the highest Corr but lower Acc-5/Acc-7 and worse

MAE than the basic pool. We therefore use the *basic* unimodal expert pool before operator-level analysis.

Based on these preliminary results, the candidate-space analysis proceeds from the *basic* unimodal expert pool, the *medium* edge-operator pool, and the *large* multimodal expert pool.

Table 15 reports the leave-one-operator-out results for the selected medium edge-operator pool.

Remove	Acc-7↑	Acc-5↑	Acc-2↑	F1↑	MAE↓	Corr↑
TemporalDiff	**54.16**	**55.84**	**85.48**	**85.48**	**0.534**	**0.769**
dwconv3	53.88	55.73	85.14	85.14	0.535	0.766
TemporalShift	53.75	55.58	85.27	85.27	0.535	0.768
full	53.69	55.65	84.84	84.84	0.537	0.766
Identity	53.41	55.43	84.63	84.63	0.542	0.767
Conv1d(k=5)	53.02	55.01	84.21	84.20	0.543	0.764
Conv1d+GELU	52.74	54.72	83.96	83.96	0.546	0.762
Conv1d(k=3)	52.81	54.98	83.47	83.47	0.545	0.764
Linear	51.48	53.56	82.98	82.98	0.553	0.756
MaxPool1d	50.86	53.18	82.56	82.56	0.550	0.754
ChannelAttn	50.67	52.89	82.13	82.13	0.559	0.751

Table 15: Leave-one-operator-out ablation on the selected medium edge-operator pool. "Remove" indicates the operator excluded from the candidate pool. Best values are highlighted in **bold**.

The results in Table 15 reveal an uneven contribution of the candidate edge operators. Removing `ChannelAttn`, `Linear`, or `MaxPool1d` leads to pronounced degradation across multiple metrics. In contrast, removing `TemporalDiff`, `TemporalShift`, or `dwconv3` does not decrease the reported performance in this leave-one-operator-out setting.

Table 16 presents the corresponding operator-level ablation for the selected basic unimodal expert pool.

Remove	Acc-7↑	Acc-5↑	Acc-2↑	F1↑	MAE↓	Corr↑
Full candidate pool	**53.69**	55.65	84.84	84.84	**0.537**	**0.766**
MHA	52.69	55.76	**85.26**	**85.27**	0.540	0.764
dwconv3	52.95	**56.03**	84.64	84.64	0.543	0.753
linear+glu	52.35	55.26	84.73	84.73	0.542	0.760
lstm	53.02	**56.03**	81.34	81.34	0.543	0.760
ChannelAttn	52.58	55.51	84.46	84.46	0.545	0.759
self_attn	52.16	55.21	84.62	84.62	0.545	0.758
transformer	50.91	53.47	82.86	82.86	0.551	0.754

Table 16: Leave-one-operator-out ablation on the selected basic unimodal expert pool. "Remove" indicates the expert type excluded from the candidate pool. Best values are highlighted in **bold**.

From Table 16, removing `transformer` causes the largest overall degradation across several metrics, whereas removing `lstm` produces a particularly large decrease in Acc-2/F1. Removing `MHA` does not uniformly decrease all classification metrics, although MAE, Corr, and Acc-7 are worse than those of the full candidate pool. The remaining operators produce smaller but measurable changes.

Finally, Table 17 reports the leave-one-operator-out ablation for the selected large multimodal expert pool.

The multimodal results show that removing `self_attn` improves all reported metrics in this leave-one-operator-out experiment. In contrast, removing `sum`, `concat`, or `MHA` produces larger degradation, indicating that these fusion primitives contribute substantially to the candidate space.

It is important to distinguish the exploratory candidate-space model from the final STREAM model. The former evaluates an unpruned candidate pool, whereas the latter uses the refined operator set obtained after the operator-removal analysis. Concretely, the final configuration used for the main STREAM results adopts the *basic* unimodal expert pool, the *medium* edge-operator pool with `TemporalShift` and `TemporalDiff` removed, and the *large* multimodal expert pool with `self_attn` removed.

Remove	**Acc-7↑**	**Acc-5↑**	**Acc-2↑**	**F1↑**	**MAE↓**	**Corr↑**
self_attn	**54.23**	**55.91**	**85.32**	**85.31**	**0.534**	**0.769**
Full candidate pool	53.69	55.65	84.84	84.84	0.537	0.766
CrossAttention	53.38	55.39	84.46	84.46	0.540	0.765
linear	53.02	55.06	84.08	84.08	0.543	0.763
linear+tanh	52.96	54.93	84.11	84.11	0.541	0.764
sum	53.24	52.17	83.24	83.25	0.549	0.758
concat	52.11	53.88	83.01	83.01	0.551	0.759
MHA	52.57	53.92	82.67	82.67	0.554	0.755

Table 17: Leave-one-operator-out ablation on the selected large multimodal expert pool. "Remove" indicates the operator excluded from the candidate pool. Best values are highlighted in **bold**.

F.2 Loss Weight Sensitivity

We further investigate the sensitivity of STREAM to the balancing coefficients of the PID-inspired decoupling loss and the DCCA alignment loss on CMU-MOSEI. As shown in Table 18, the model exhibits relatively stable performance across different choices. Among the evaluated configurations, $\lambda_{\text{pid}} = 0.2$ and $\lambda_{\text{dcca}} = 0.1$ produce the reported STREAM result and are therefore used as the default setting in the main experiments.

λ_{pid}	λ_{dcca}	Acc-7↑	Acc-5↑	Acc-2↑	F1↑	MAE↓	Corr↑
0.1	0.01	54.09	55.68	84.90	84.90	0.536	0.755
0.1	0.05	53.81	55.39	85.31	85.31	0.532	0.758
0.1	0.10	52.87	54.73	84.31	84.31	0.547	0.748
0.2	0.01	54.18	55.75	85.26	85.26	0.533	0.757
0.2	0.05	54.02	55.52	85.55	85.55	0.538	0.753
0.2	**0.10**	**54.58**	**56.10**	**85.74**	**85.73**	**0.531**	**0.772**
0.5	0.01	54.22	55.79	84.84	84.84	0.538	0.757
0.5	0.05	53.76	55.54	85.17	85.18	0.536	0.756
0.5	0.10	53.93	55.41	85.67	85.67	0.540	0.755

Table 18: Parameter sensitivity analysis of STREAM on CMU-MOSEI under different loss balancing coefficients. We vary the weights of the PID-inspired decoupling loss (λ_{pid}) and the DCCA alignment loss (λ_{dcca}). The reported default configuration is highlighted in bold.

F.3 Expert Usage Analysis

To further examine the routing behavior, we provide diagnostics from two complementary perspectives. First, we measure the agreement between the learned routing gates and post-hoc attribution scores. Second, we summarize the average expert usage statistics and top-1 selection frequency at the low-level and high-level branches.

In Tables 19 and 20, `Avg_Gate` and `Avg_Attr` denote the mean routing weight and mean post-hoc attribution score, respectively. T1R and T1A denote the proportions of samples for which an expert is ranked first by the routing gate and attribution score, respectively. The suffix Uni denotes a modality-specific Uniqueness Expert.

Level	**Expert**	**Avg_Gate**	**Avg_Attr**	**T1R**	**T1A**
Low	Text-Uni	0.103	0.112	0.108	0.116
Low	Audio-Uni	0.020	0.018	0.019	0.018
Low	Vision-Uni	0.576	0.548	0.594	0.603
Low	Synergy	0.228	0.239	0.214	0.197
Low	Redundancy	0.073	0.083	0.065	0.066

Table 19: Expert usage statistics of the low-level branch on CMU-MOSEI. **Avg_Gate** and **Avg_Attr** denote the mean routing weight and mean attribution score of each expert, respectively. **T1R** and **T1A** denote the proportion of samples in which the corresponding expert is ranked first by routing gates or attribution scores.

As shown in Table 19, the low-level branch exhibits an uneven but non-degenerate usage distribution. The Vision-Uni Expert has the largest average routing weight and the highest top-1 frequency, while

the Synergy Expert is the second most active expert. Text-Uni and Redundancy contribute more moderately, and Audio-Uni receives the lowest average weight. The non-zero usage of all five expert types indicates that the branch retains multiple active pathways despite its strong visual preference.

Level	Expert	Avg_Gate	Avg_Attr	T1R	T1A
High	Text-Uni	0.649	0.664	0.742	0.754
High	Audio-Uni	0.048	0.043	0.026	0.027
High	Vision-Uni	0.079	0.078	0.050	0.051
High	Synergy	0.092	0.097	0.063	0.062
High	Redundancy	0.132	0.118	0.119	0.106

Table 20: Expert usage statistics of the high-level branch on CMU-MOSEI. **Avg_Gate** and **Avg_Attr** denote the mean routing weight and mean attribution score of each expert, respectively. **T1R** and **T1A** denote the proportion of samples in which the corresponding expert is ranked first by routing gates or attribution scores.

A different usage pattern emerges in the high-level branch. As reported in Table 20, the Text-Uni Expert has the largest average routing weight and top-1 selection frequency. The Redundancy Expert is the second most prominent component, while the Synergy and Vision-Uni Experts retain smaller but non-zero contributions.

For the consistency metrics in Table 21, T1M denotes the proportion of samples for which the top-1 expert under the routing gate agrees with the top-1 expert under the post-hoc attribution ranking, while T2O denotes the average Jaccard overlap between the corresponding top-2 expert sets.

Level	Pearson↑	Spearman↑	T1M↑	T2O↑
Low	0.756	0.731	0.807	0.836
High	0.832	0.806	0.861	0.879
Avg	0.794	0.769	0.834	0.858

Table 21: Consistency between routing gates and post-hoc attribution scores on CMU-MOSEI. **T1M** denotes the proportion of samples for which the top-1 expert selected by the routing gate coincides with the top-1 expert ranked by attribution. **T2O** denotes the average Jaccard overlap between the top-2 expert sets induced by routing gates and attribution scores.

Taken together, Tables 19– 21 show substantial agreement between routing weights and post-hoc attribution scores while also revealing a level-specific division of labor. The low-level branch places greater routing mass on Vision-Uni and Synergy, whereas the high-level branch places substantially greater routing mass on Text-Uni and a secondary amount on Redundancy.

G Dataset Descriptions

We evaluate STREAM on three widely used multimodal sentiment analysis benchmarks, including two English datasets, CMU-MOSI and CMU-MOSEI, and one Chinese dataset, CH-SIMS v2.0. All three datasets contain textual, acoustic, and visual information and are used to evaluate the model under different dataset scales, languages, and multimodal interaction patterns.

CMU-MOSI. CMU-MOSI is an English multimodal sentiment analysis dataset collected from YouTube opinion videos. It contains 2,199 opinion-level segments extracted from 93 videos, with each segment annotated by a sentiment intensity score ranging from -3 (strongly negative) to $+3$ (strongly positive). The dataset provides textual, acoustic, and visual modalities and is commonly used for multimodal sentiment regression and classification. The official dataset resources are available at `https://github.com/CMU-MultiComp-Lab/CMU-MultimodalSDK`.

CMU-MOSEI. CMU-MOSEI is a larger English benchmark for multimodal sentiment and emotion analysis in real-world online videos. It contains 23,453 sentence-level video segments from more than 1,000 speakers and covers over 65 hours of video content across a wide range of topics. Each segment is accompanied by textual, acoustic, and visual information and is annotated with sentiment intensity as well as multiple emotion categories. The official dataset information is available at `https://cmu-multicomp-lab.github.io/multibench/datasets/`.

CH-SIMS v2.0. CH-SIMS v2.0 is a Chinese multimodal sentiment analysis dataset designed to place greater emphasis on acoustic and visual sentiment cues. It contains 14,563 video segments collected from 11 diverse real-world scenarios, including 4,402 supervised samples and 10,161 unlabeled samples. The supervised portion provides multimodal sentiment labels together with modality-specific annotations. The official project page is available at `https://thuiar.github.io/sims.github.io/chsims`.

H Reproducibility Guide

This section describes the implementation structure used to reproduce the experimental pipeline of STREAM. The notation follows the main paper and Algorithm 1.

H.1 Environment Configuration

STREAM is implemented in Python using PyTorch and is trained on a CUDA-enabled graphics processing unit. Our reference environment is summarized below:

- Operating system: Ubuntu 20.04 or a compatible Linux distribution;
- Python: 3.10;
- PyTorch: 2.0.1;
- CUDA: 11.7.

H.2 Input Data Format

We use pre-extracted textual, acoustic, and visual features. Each dataset is stored as a Python dictionary containing the training, validation, and test partitions:

```
dataset = {
    "train": {
        "text": ...,
        "audio": ...,
        "vision": ...,
        "regression_labels": ...,
        "id": ...
    },
    "valid": {
        "text": ...,
        "audio": ...,
        "vision": ...,
        "regression_labels": ...,
        "id": ...
    },
    "test": {
        "text": ...,
        "audio": ...,
        "vision": ...,
        "regression_labels": ...,
        "id": ...
    }
}
```

The modality tensors follow $\mathbf{X}^m \in \mathbb{R}^{N\times T\times d_m}$, where N denotes the number of samples, T the aligned sequence length, and d_m the input dimensionality of modality $m \in \{t, a, v\}$. The regression labels are continuous sentiment scores.

H.3 Data Loading

The following simplified example illustrates the expected loading procedure:

```
with open(data_path, "rb") as f:
    dataset = pickle.load(f)

def load_split(split):
    text = torch.tensor(
        dataset[split]["text"],
        dtype=torch.float32
    )
    audio = torch.tensor(
        dataset[split]["audio"],
        dtype=torch.float32
    )
    vision = torch.tensor(
        dataset[split]["vision"],
        dtype=torch.float32
    )
    labels = torch.tensor(
        dataset[split]["regression_labels"],
        dtype=torch.float32
    ).view(-1)

    return text, audio, vision, labels
```

The model is trained using the continuous regression labels. Discrete classification labels are generated only during evaluation from the ground-truth sentiment values and the continuous model predictions, according to the evaluation mappings described below. They are therefore not separate training targets and do not require an independent classification head.

The training loader uses random shuffling and drops the final incomplete mini-batch. The validation and test loaders preserve sample order and do not drop samples.

H.4 Representative Forward Pass

The following pseudocode follows the same notation and computational order as the main method and Algorithm 1.

```
def forward(text, audio, vision):

    projected = {
        "text": text_projection(text),
        "audio": audio_projection(audio),
        "vision": vision_projection(vision)
    }

    X_m_LL, X_m_HL = hierarchical_extractor(projected)

    X_LL, low_aux = low_level_apmoe(X_m_LL)
    X_HL, high_aux = high_level_apmoe(X_m_HL)

    Z_L, Z_H, loss_dcca = dcca_alignment(
        X_LL,
        X_HL
    )

    X_LL_out, X_HL_out = collaborative_bimamba(
        Z_L,
        Z_H
    )

    representation = pool(
        X_LL_out,
        X_HL_out
    )
    pred_reg = regression_head(representation)

    return (
        pred_reg,
        low_aux,
        high_aux,
        loss_dcca
    )
```

Here, `X_m_LL` and `X_m_HL` correspond to the modality-wise hierarchical streams $\{\mathbf{X}_m^{\mathrm{LL}}\}$ and $\{\mathbf{X}_m^{\mathrm{HL}}\}$ in the main paper. `X_LL` and `X_HL` correspond to the fused outputs $\mathbf{X}_{\mathrm{LL}}$ and $\mathbf{X}_{\mathrm{HL}}$ of L-APMoE and H-APMoE.

The auxiliary outputs `low_aux` and `high_aux` contain the full-view and modality-perturbed $\mathbf{Y}^u$, $\mathbf{Y}^s$, and $\mathbf{Y}^r$ representations used by the PID-inspired objective. They are not the representations supplied directly to DCCA.

DCCA operates on $\mathbf{X}_{\mathrm{LL}}$ and $\mathbf{X}_{\mathrm{HL}}$ and produces the aligned latent representations $\mathbf{Z}_{\mathrm{L}}$ and $\mathbf{Z}_{\mathrm{H}}$. These two representations are subsequently processed by Collaborative Bi-Mamba. The final prediction is a single continuous sentiment value. No separate classification output is generated.

H.5 Optimization Procedure

We use AdamW with a One-Cycle learning-rate schedule. Projection, hierarchical-extraction, and routing modules use a reduced maximum learning rate of 10^{-4}, while the Mamba fusion layers and prediction heads use 10^{-3}. The warm-up portion occupies 15% of the total optimization steps. Unless otherwise specified, the loss coefficients are set to $\lambda_{\mathrm{pid}} = 0.2$ and $\lambda_{\mathrm{dcca}} = 0.1$, consistent with Table 18.

Mixed-precision training is enabled using automatic mixed precision. Before each optimizer step, the gradients are unscaled and clipped to a maximum norm of 1.0. A simplified training iteration consistent with Eq. equation 5 and Algorithm 1 is given below:

```
optimizer.zero_grad()

with torch.cuda.amp.autocast():
    pred_reg, low_aux, high_aux, loss_dcca = model(
        text,
        audio,
        vision
    )

    loss_reg = huber_loss(
        pred_reg,
        labels
    )

    # Each PID loss is computed from the corresponding
    # full-view and masked-view Y^u, Y^s, and Y^r outputs.
    loss_pid_low = pid_loss(low_aux)
    loss_pid_high = pid_loss(high_aux)
    loss_pid = loss_pid_low + loss_pid_high

    loss = (
        loss_reg
        + 0.2 * loss_pid
        + 0.1 * loss_dcca
    )

scaler.scale(loss).backward()
scaler.unscale_(optimizer)
clip_grad_norm_(model.parameters(), 1.0)
scaler.step(optimizer)
scaler.update()
scheduler.step()
```

The optimization objective is therefore exactly

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{reg}} + \lambda_{\mathrm{pid}} \left(\mathcal{L}_{\mathrm{pid}}^{\mathrm{LL}} + \mathcal{L}_{\mathrm{pid}}^{\mathrm{HL}}\right) + \lambda_{\mathrm{dcca}} \mathcal{L}_{\mathrm{dcca}}, \tag{9}$$

which is equivalent to Eq. equation 5 in the main paper after defining

$$\mathcal{L}_{\mathrm{pid}} = \mathcal{L}_{\mathrm{pid}}^{\mathrm{LL}} + \mathcal{L}_{\mathrm{pid}}^{\mathrm{HL}}. \tag{10}$$

For each semantic level,

$$\mathcal{L}_{\mathrm{pid}}^{(\cdot)} = \mathcal{L}_{\mathrm{uni}}^{(\cdot)} + \mathcal{L}_{\mathrm{syn}}^{(\cdot)} + \mathcal{L}_{\mathrm{red}}^{(\cdot)}, \tag{11}$$

where $(\cdot) \in \{\mathrm{LL}, \mathrm{HL}\}$ and the three constituent terms are defined exactly as in Sec. 3.4 of the main paper.

H.6 Evaluation Protocol

The test set is evaluated only after model selection has been completed on the validation set. STREAM outputs only the continuous regression prediction $\hat{y}$. We report mean absolute error and Pearson correlation directly from this continuous prediction.

Classification metrics are derived post hoc from $\hat{y}$ using the following mappings:

$$\hat{y}_7 = \text{clip}\left(\text{round}(\hat{y}), -3, 3\right), \tag{12}$$

$$\hat{y}_5 = \text{clip}\left(\text{round}(\hat{y}), -2, 2\right), \tag{13}$$

$$\hat{y}_2 = \mathbb{I}[\hat{y} \geq 0]. \tag{14}$$

The corresponding ground-truth class labels are derived from the continuous ground-truth sentiment values using the same evaluation protocol. Acc-7 and Acc-5 are computed from the corresponding discretized labels. Binary accuracy and weighted F1 use zero as the positive–negative decision boundary. Pearson correlation is computed directly from the continuous predictions and continuous ground-truth sentiment scores.

Thus, the classification metrics reported in the main paper and supplementary material are evaluation views of the regression output rather than outputs of a separately trained classification head.

H.7 Correspondence Between Mathematical and Implementation Notation

For clarity, Table 22 summarizes the notation used consistently across the main paper, Algorithm 1, and the implementation pseudocode.

Main-paper notation	Implementation notation	Meaning
$\mathbf{X}_m^{\mathrm{LL}}$	`X_m_LL`	Low-level representation of modality m before L-APMoE
$\mathbf{X}_m^{\mathrm{HL}}$	`X_m_HL`	High-level representation of modality m before H-APMoE
$\mathbf{X}_{\mathrm{LL}}$	`X_LL`	Fused output of L-APMoE
$\mathbf{X}_{\mathrm{HL}}$	`X_HL`	Fused output of H-APMoE
$\mathbf{Y}^u$	`Y_u`	Uniqueness Expert representation used by the PID-inspired objective
$\mathbf{Y}^s$	`Y_s`	Synergy Expert representation used by the PID-inspired objective
$\mathbf{Y}^r$	`Y_r`	Redundancy Expert representation used by the PID-inspired objective
$\mathbf{Z}_{\mathrm{L}}$	`Z_L`	DCCA-projected low-level representation
$\mathbf{Z}_{\mathrm{H}}$	`Z_H`	DCCA-projected high-level representation
$\mathbf{X}_{\mathrm{LL}}^{\mathrm{out}}$	`X_LL_out`	Collaborative Bi-Mamba output for the low-level stream
$\mathbf{X}_{\mathrm{HL}}^{\mathrm{out}}$	`X_HL_out`	Collaborative Bi-Mamba output for the high-level stream
$\hat{y}$	`pred_reg`	Continuous sentiment prediction

Table 22: Correspondence between the mathematical notation in the main paper and the implementation notation used in the supplementary pseudocode.

H.8 Availability Statement

The complete implementation includes dataset-specific preprocessing, configuration files, checkpoint management, routing-analysis utilities, and scripts for reproducing all tables and figures. To preserve anonymous review, the repository is not linked in the current submission. The complete code and documentation will be made publicly available upon acceptance.