

# Controlled Case Studies of Three Add-on Families in Frozen-CLIP Multimodal Sarcasm Detection

Ruochong Gu | July 2026

## Abstract

Multimodal sarcasm systems usually change several different modules at the same time. It would be difficult when we want to find the value that each module provide, since the value of one single module can't be represented by the whole system. In this reasearch, same testing environments are used to test the value of three different modules that are used in multimodal sarcasm systems. We found that including these modules will increase the scores on some testing indexes, but may also decrease the points on other indexes. We also found that although there are improvements, the magnitude are rather slight. Results show that the performance of multimodal sarcasm system are usually the result of a combined effect.

## 1. Introduction

Multimodal sarcasm often depends on the relationship between the text and image, rather than just base on one single modal like only text or only image. When a text is paired with a unsuitable image, sarcasm forms. There are three main interventions, including shared or private decomposition (D), external knowledge generation (K) and robust training (R), representing methods used in framework design, information assist and training process. Since most systems now are different in either encoders and objectives or prompts and benchmarks, directly compairing them would be impossible to show the value on each single module.

This research compairs the three methods mentioned above based on MMSD2.0, and all configuration are strictly the same, based on OpenCLIP ViT-B/32 cache with OpenAI weights, seeds 42 to 51, and the same evaluation code and training process.

The study asks three narrowly defined research questions:

- Question 1: decomposition configuration. Under matched frozen features and training, dose the decomposition framework really works? Where does the value come from?
- Question 2: external generated knowledge. Dose the external knowledges really improves the performance of the system alone?
- Question 3: robustness What is the benefit and price of implementing a robust training?

The study contributes an auditable comparison workflow that combines a shared protocol with seed-level and per-sample evidence.

## 2. Literature Review

### 2.1 Shared/Private Factorization and Cross Modal Interaction

This method focus on treating the multimodal prediction in a way of seperating the two modals and them model them with specific method, which can help to better deal with the relationship and contradiction between modals, and finally improve the performance. MInD uses one sharerd encoder and two private encoders, splitting the informations into shared and private groups. CoDe uses semantic complement and characteristic decomposion to solve problems like emotion inconsistency. D&R Net will first explicitly

seperates what is shared and what is private between modals, while DMMD split the characteristic space into irrelevant and specific subspaces, and also included multi-grained distillation to improve performance.

The nature of sarcasm is inconsistensy, so the method should not be limited with characteristic decomposition, but to model the inconsistency between text and image through different grain size and different levels. MICL models the inconsistency through three grain sizes at the same time, including token and patch, entity and object, and sentiment views. AMIF learns the inconsistency through text level, image level and cross modal level, and automatically weights them according to the sample's degree of ambiguity. ITFNet prioritize the inconsistency and contradiction between modals rather than persueing sentiment consistency as a whole, and builds fact and sentiment based fields.

These studies showed that different models will improve the performance, but we can't tell whether the improvements really came from seperating the characteristic and compairing them, or from better encoders or lost function. The D family extracts the core idea of 'shared or private decomposition', and applies a shared or private head to a fixed environment.

## 2.2 LVLM-Generated Context

A second method is to introduce external context into the text and image pair, since the establishment of sarcasm relies on external knowledge and commonsense, which is not include in the text, or couldn't be seen in the image. WisdoM uses LVLM to generate relative knowledge, them applies a fusion mechanism to reduce noise. GDCNet generates fact-based image description, and since the description can provide more accurate sentiment standard, models can better grasp the difference between the image and text. SemIRNet follows a knowledge base approach, calling ConceptNet and calculating cross modal similarity, providing the model with common knowledge.

Although benefits are obvious, LVLM generated knoledge also has to be treated cautiously. Wang, Zhang and Jing reported that under a zero-shot environment, the ability of visual understanding and abstract content understanding has dignificant weakness, meaning that knowledge genereated by LVLM may not be accurate and closly related. MiDRE designed a learned gate, mixing the text-image expert and LVLM expert, and determain the context weight by the gate. However, the gate can only reduce the impact of low quality knowledge, they can still have a negative impact on the system.

The K family studies uses LVLM to generate visual description and background knowledge, making sure that it doesn't guess answer directly. It's seperated into different groups, including no context, learnd zero-context, disordered context, and aligned context, to compare the value of the generated knowledge.

## 2.3 Missing-Modality Robustness and Deployment Alternatives

In real conditions, lost of modal, low quality modal and fake association are possible, so robust training should be included. DRF is a typical active recovery method, through pre-estimating the feature distributions, the system can reduce the weight of the low quality or lost modal, to improve stability. MCIB focus on dealing with the problem of shortcut learning, which forces the model to learn the true cross-modal relationshihp, raather than being distracted by one single modal noise or inaccuracy. InterCLIP-MEP inject cross modal information into the CLIP encoder to strengthen the interaction between modals, and also includes a dynamic memory enhanced predictor, increasing the reliability when information is insufficient.

By implementing the robust training, models learns to deal with lost information. The R family studies can help to determain how robust training works on both clear data and data that has lost information.

## 2.4 Controlled Comparison Gap

The existed literature provides acceptable reason for models to implement decomposition, external knowledge generation and robust training, but since they all work on different environments, there is still a comparison gap. In one system, representation structure, knowledge source, fusion method and training objective may be all changed, so it is impossible to directly find out whether one specific module could improve the performance. This study uses a frozen-OpenCLIP resource, where only one configuration or internal controls are changed at one time. All the tested D, K and R methods are simplified for controlled testing, so the results may be different to previous studies.

Table 1. Synthesis of the closest literature and the present comparison gap.

| Study         | Primary idea                                                         | Evaluation emphasis                                              | Why not directly comparable                                                                     |
|---------------|----------------------------------------------------------------------|------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| D&R Net       | Shared/unique decomposition plus a relation network                  | Task-specific multimodal sarcasm model                           | Changes encoder, decomposition, and relation architecture together                              |
| DMMD          | Modality-agnostic/specific subspaces with multi-grained distillation | Disentangled sarcasm representation                              | Uses distillation objectives and a different end-to-end design                                  |
| Tang et al.   | Visual instruction and retrieved demonstrations for a generative LLM | MMSD2.0 plus OOD RedEval                                         | Requests task prediction rather than without task judgments auxiliary context                   |
| AMIF / ITFNet | Multi-level ambiguity or fact/sentiment tension for incongruity      | Fine-grained discrepancy modeling                                | Adds specialized task structures without matched capacity/mask controls                         |
| MCIB          | Conditional information bottleneck against shortcut learning         | Reconstructed MUSTARD++R                                         | Different modality format, benchmark, and bottleneck objective                                  |
| Present study | Three frozen-feature family case studies with internal controls      | One official MMSD2.0 split; ten seeds and condition stress tests | Controls expose but do not eliminate all capacity, topology, and external-validity alternatives |

The table compares research questions and experimental roles rather than published scores, because the source studies do not share a common backbone, training protocol, or deployment condition.

## 3. Methodology and Experimental Framework

### 3.1 Research Design

The study distinguishes protocol-frozen, derived, and exploratory analyses; Table 2 summarizes their roles. Because exploratory V1 had already used the official test split, V2 intervals are descriptive rather than test-blind confirmation.

Table 2. Formal configurations, comparison roles, and evidence tiers.

| Configuration family           | Configurations                                       | Comparison role                                                                                 |
|--------------------------------|------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Base controls                  | B1–B5                                                | Protocol-frozen supportive modality and partial feature/capacity controls; no equal-capacity B3 |
| Shared/private factorized head | D1–D4                                                | D1–B4 primary; D1–B5 and loss ablations supportive; topology cause unresolved                   |
| Context                        | K0–K3                                                | K3–K1 primary; K3–K2 supportive alignment control                                               |
| Robustness                     | B4 versus R1                                         | Protocol-frozen primary clean, missing, and noise conditions                                    |
| Sensitivity                    | B4/D1/K3/R1                                          | Exploratory validation-only tuning; Appendix B                                                  |
| Derived controls               | Routing; late fusion; exact text; two-way resampling | Post-review derived deployment and sensitivity analyses                                         |

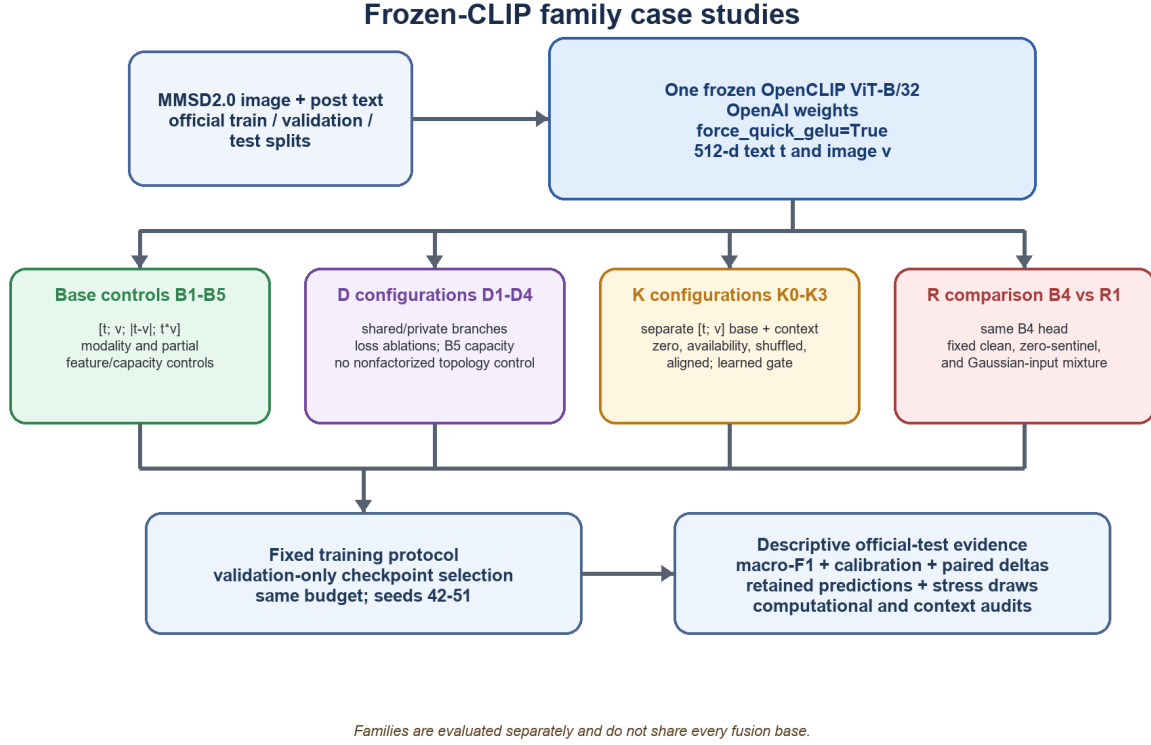


Figure 1. Frozen-feature architecture and comparison logic. The D, K, and R families are separate case studies and do not share every fusion base.

### 3.2 Dataset

MMSD2.0 is the only benchmark used for formal analysis. It contains 24,635 image-text pairs divided into 19,816 training, 2,410 validation, and 2,409 test samples. The test set contains 1,372 non-sarcastic and 1,037 sarcastic samples.

To make sure that the training samples are exactly different from the testing samples, an exact-string sensitivity analysis is used to compare the samples, and recomputes the principal contrasts on the 2,326 unseen texts. The matches are then grouped by the training label relation. This check detects literal repetition, not paraphrases or near duplicates.

### 3.3 Shared Features, Base Model, and Training

All the configurations will use the same cache, containing a 512 dimensional feature of text and image generated from OpenCLIP ViT-B/32 with OpenAI weights preprocessed with OpenAI transformer and tokenizer, along with a forced uniform activation function (`force_quick_gelu=True`). The encoders are frozen through out the training. We first assume that the L2 normalized text feature is  $t$  and image feature is  $v$ . The B family is the baseline of the model, with no extra modules introduced. There are five baselines in all. B1 and B2 are pure text and image respectively, B3 is a simple joint of text and image. B4 is a more complex standard fusion. It is a joint of 4 features with a two layer MLP and 256 dimensional layer, matched with GELU, LayerNorm, 0.2 dropout. The final output will be a two-class output. B5 is a baseline that has 320 dimensions and 656,962 parameters, which is close to the D family, to eliminate disturbance term.

$$z\_B = [t; v; |t - v|; t \odot v], \quad \text{logits\_B} = \text{MLP\_B}(z\_B).$$

B3 and B4 contain 263,426 and 525,570 trainable parameters, respectively. B5 on the other hand is closer to D family in parameters, at 656,962 (D1 is 658,818). Appendix A provides a more precise accounting.

The training configuration are fixed. AdamW optimizer with learning rate 0.001, weight decay 0.0001, batch size 512 and at most 30 epochs. Training will stop early if the macro-F1 score doesn't improve for 5 consecutive epochs. The one with highest macro-F1 score will be selected as the final score. More analysis is reported in Appendix B.

Another late-fusion model are also set as control, which combines B1 and B2 probabilities.

$$p = \alpha p_{B1} + (1-\alpha)p_{B2}.$$

Text weight  $\alpha$  will be selected from  $\{0.00, 0.05, \dots, 1.00\}$ , according to the macro-F1 score. If more than one  $\alpha$  share a similar score, then prioritize the one closer to 0.5 and then the smaller  $\alpha$ .  $\alpha$  is frozen before test evaluation.

### 3.4 Shared/Private Factorized Head (+D)

There are 4 configurations in the D family. D1 is the complete version, containing feature encoding layer, reconstruct and decoding layer and classification output layer. It has two encoders: weight-shared encoder and modal private encoder. The weight-shared encoder use one encoder to process both text and image at the same time, forcing it to extract information that are cross modal consistent. The modal private encoder on the other hand, uses two separate encoders to extract unique information from each modal. D2 removes orthogonality, D3 removes reconstruction and D4 removes both.

$$s_t = E_s(t), s_v = E_s(v), p_t = E_t(t), p_v = E_v(v).$$

The lost function for D1 is as follow.

$$L_D = L_{CE} + 0.1L_{orth} + 0.1L_{rec}.$$

The  $L_{CE}$  is Cross-entropy loss,  $L_{orth}$  is Orthogonal loss and  $L_{rec}$  is Reconstruction loss. During the Ablation study, the structure of the lost function is not changed; only the weight of each loss are changed. For example, in D2, the weight of orthogonality loss are set to 0.

### 3.5 Dedicated LVLM-Generated Context Family (+K)

In this session, I used glm-4.1v-thinking-flash API to generate visual facts, text context and background knowledge. The prompt prohibited the model from directly judge sarcasm, emotional polarity, implicit intent inference and so on, that would be a potential for the LVLM model to guess the answer out, making sure that it is always only an assist for the system to judge. During the generating process, there are 24,026 complete records, 609 API failures and 185 records marked as unavailable due to the result containing target judgement words that may affect the system, leaving 23,841 available records. All 794 unavailable records are marked as zero context vector and availability  $m = 0$ .

Each usable context will be encoded separately by the OpenCLIP text encoder same as the baseline. First, visual facts, text context and background knowledge will be separately encoded and perform L2-normalization. Then, calculate the average of the three normalized vectors, and perform L2-normalization again to get the final context representation vector  $k$ . The context fusion structure adopted a structure of “base branch + context branch”, along with a self-learning gate to automatically control the context injection. In the formula,  $h_b$  is basic representation,  $k_k$  is context representation,  $m$  is availability and  $g$  is gating.

$$k = \text{norm}((1/3)\sum_j \text{norm}(e(q_j))); \quad g = \sigma(W_g[h_b; k_k; m]); \quad h = h_b + m(g \odot h_k).$$

There are 4 configurations in the K family. K0 is  $k = 0$  and  $m = 0$ ; K1 is  $k = 0$  and real  $m$ ; K2 is random arranged context and real  $m$ ; and K4 is sequential arranged context and real  $m$ . this forms a contrast experiment that can clearly show how the generatex context is affecting the system.

### 3.6 Zero-Vector Missingness Adaptation (+R)

The structure and scale of R1 is the same as B4, the only difference is how the data is distributed when training, making sure that all the performance gap are due to the difference in training. R1 is structured as 50% clean samples, 15% missing image samples, 15% missing text samples, 10% samples adding 0.1 image noise and 10% samples adding 0.1 text noise. The gaussian noise is calculated as follows:

$$x' = x + \frac{\sigma}{\sqrt{512}}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

The features after adding noise will undergo a same normalization process same to the original ones, making sure that the inputs are consistent. Same as Baselines, the checkpoints will also be chosen from the macro-F1 score with the clean test sets. The validation process will be assessing the model through 7 different input conditions: clean inputs, image missing, text missing, image noise 0.1, image noise 0.2, text noise 0.1, text noise 0.2. to avoid random fluctuation, the model will be tested under the same set of sample for 10 times and calculate their average score.

Moreover, another “availability-aware route” method is also tested, which is closer to actual conditions. When all data are present, multimodal model will be used; when either image or text modal are lost, a singlemodal model will be used.

### 3.7 Comparison Hierarchy, Metrics, and Computational Audit

Since the test sets have a slight imbalance between sarcasm and non-sarcasm samples, where non-sarcasm samples are more, directly comparing accuracy will not be appropriate. Thus, macro-F1 are introduced to be the main standard, since it calculates the average F1 score for both kinds and sets a same weight on the two scores, it can reflect the model’s ability more accurately. Also, all the comparison within each family reports 5 contents: values, means and sample standard deviation for the seeds, paired difference, unadjusted 95% t intervals, positive seed conds and an exact two-sided McNemar test with a Holm adjustment. This further ensures the completeness or information and fairness of comparison. Paired two-way-seeds are also used to increase reliability. It will consider the randomness in training seeds and the randomness of tset samples. Through 5000 repetitions, each time will simultaneously perform sampling with replacement on both training seeds and test samples. After that, macro-F1 are calculated again and save the average difference for the pair, and the 2.5<sup>th</sup> and 97.5<sup>th</sup> percentiles will be used to form a 95% sensitive interval.

### 3.8 Reproducibility Environment, Context Configuration, and Provenance

The training runs on a server with Tesla V100-SXM2-32GB GPU and Python 3.12.3, PyTorch 2.8.0+cu128, CUDA runtime 12.8, cuDNN 9.10.2, open\_clip\_torch 3.3.0 and NumPy 2.3.2. For each random seed, Python, NumPy, PyTorch CPU, and all CUDA generators were set to the same seed, and CUBLAS\_WORKSPACE\_CONFIG=:4096:8, cudnn.benchmark=False, cudnn.deterministic=True, and torch.use\_deterministic\_algorithms(True, warn\_only=True) were set.

For the context generation part, 4 worker threads were configured, with 1sec interval, 120sec timeout and up to 12 API request retries for each sample. If the generated contents include target language, there are up to 3 corrective generations. The request configuration is set to max\_tokens = 4096 and no label included.

## 4. Results

### 4.1 Base and Modality Controls

Table 3. MMSD2.0 clean test overview: formal configurations and derived late-fusion control (mean  $\pm$  sample SD over seeds 42-51)

| Method      | Accuracy            | Macro-F1            | F1 non-sarcastic    | F1 sarcastic        |
|-------------|---------------------|---------------------|---------------------|---------------------|
| B1          | 0.8008 $\pm$ 0.0033 | 0.7998 $\pm$ 0.0032 | 0.8141 $\pm$ 0.0041 | 0.7855 $\pm$ 0.0025 |
| B2          | 0.7848 $\pm$ 0.0023 | 0.7835 $\pm$ 0.0024 | 0.8004 $\pm$ 0.0030 | 0.7666 $\pm$ 0.0042 |
| B3          | 0.8356 $\pm$ 0.0039 | 0.8346 $\pm$ 0.0037 | 0.8473 $\pm$ 0.0048 | 0.8219 $\pm$ 0.0028 |
| B4          | 0.8383 $\pm$ 0.0016 | 0.8372 $\pm$ 0.0017 | 0.8502 $\pm$ 0.0019 | 0.8242 $\pm$ 0.0029 |
| B5          | 0.8374 $\pm$ 0.0037 | 0.8364 $\pm$ 0.0037 | 0.8495 $\pm$ 0.0043 | 0.8233 $\pm$ 0.0040 |
| D1          | 0.8430 $\pm$ 0.0048 | 0.8420 $\pm$ 0.0047 | 0.8547 $\pm$ 0.0057 | 0.8293 $\pm$ 0.0045 |
| D2          | 0.8411 $\pm$ 0.0055 | 0.8401 $\pm$ 0.0055 | 0.8526 $\pm$ 0.0057 | 0.8276 $\pm$ 0.0061 |
| D3          | 0.8426 $\pm$ 0.0058 | 0.8417 $\pm$ 0.0057 | 0.8538 $\pm$ 0.0063 | 0.8295 $\pm$ 0.0058 |
| D4          | 0.8415 $\pm$ 0.0063 | 0.8405 $\pm$ 0.0062 | 0.8529 $\pm$ 0.0066 | 0.8282 $\pm$ 0.0063 |
| K0          | 0.8389 $\pm$ 0.0019 | 0.8380 $\pm$ 0.0017 | 0.8498 $\pm$ 0.0026 | 0.8262 $\pm$ 0.0010 |
| K1          | 0.8402 $\pm$ 0.0013 | 0.8392 $\pm$ 0.0013 | 0.8518 $\pm$ 0.0016 | 0.8266 $\pm$ 0.0015 |
| K2          | 0.8362 $\pm$ 0.0034 | 0.8353 $\pm$ 0.0032 | 0.8475 $\pm$ 0.0046 | 0.8231 $\pm$ 0.0031 |
| K3          | 0.8375 $\pm$ 0.0024 | 0.8365 $\pm$ 0.0022 | 0.8494 $\pm$ 0.0032 | 0.8236 $\pm$ 0.0016 |
| R1          | 0.8346 $\pm$ 0.0023 | 0.8334 $\pm$ 0.0024 | 0.8471 $\pm$ 0.0026 | 0.8198 $\pm$ 0.0038 |
| Late fusion | 0.8340 $\pm$ 0.0021 | 0.8331 $\pm$ 0.0020 | 0.8456 $\pm$ 0.0026 | 0.8205 $\pm$ 0.0023 |

According to the table, B4 reaches a macro-F1 score of 0.8372, higher than the rest baseline scores. It also shows that the performance of single-modal, simple fusion and late fusion are all worse than B4, which has interactive features.

### 4.2 RQ1: Complete D Configuration

According to the Table 3, D1 reaches a macro-F1 score at 0.8420, which is slightly higher than B4. When comparing with B5, which has nearly the same training parameter as D1, the score is even higher. This means that the improvement comes from the decomposition structure, rather than higher training parameters. Within 24,090 paired-item tests, D4 corrected 699 errors B4 made, but produced 584 new errors, and the errors all distributed diffusely, not focused on some specific samples. This means that the decomposition method can only bring small improvements as a whole, but can't be treated as a method to correct specific mistakes.

### Per-seed macro-F1: B4 and D1 on the fixed MMSD2.0 test split

Each line joins the same training seed; this is descriptive, not an untouched-test confirmation.



Figure 2. Paired B4 and D1 macro-F1 values for seeds 42-51 on the fixed official test split.

#### 4.3 RQ2: Dedicated Context Family

Although K3 included generated context sequential to the sample sequence, it's macro-F1 score is still lower than K1, which ranks the highest in the K family. the 95% confidence interval lies entirely within the negative range, and only 1 seed in 10 seeds for K3 are better than K1, so the performance decrease is stable. This means that either the generated context can't provide enough useful new information, or the gate control can't perfectly filter the noise information out.

#### 4.4 RQ3: Zero-Vector Missingness Adaptation and Routing

Compared with B4, the only difference for R1 is that it is trained with samples including lost and noise samples, so the performance difference is entirely due to the robust training strategy. For clean samples, the macro-F1 score for R1 has slightly decreased 0.0038 than D4. This is a typical cost for robust training. To adapt for the lost and noise inputs, model has to give up some chance of performing better when samples are all clean samples. In scenarios including lost image or text however, the performance of R1 is significantly better than B4, meaning that robust training does improve the performance of the model when there are extreme conditions. When there are only noise on one modal, R1 didn't perform better than B4. The core improvement for R1 is not for recovering the lost modals, but rebalancing the decision distribution, which allows the model to learn to adjust the classification thresholds and avoid prediction bias, which also explained why this version doesn't work pretty well with noise samples.

Table 4. Zero-vector and Gaussian feature stress tests, plus availability-aware route (mean  $\pm$  sample SD over seeds 42-51)

| Condition        | B4 macro-F1         | R1 macro-F1 (delta vs B4)      | Availability-aware route (delta vs R1) |
|------------------|---------------------|--------------------------------|----------------------------------------|
| Clean            | 0.8372 $\pm$ 0.0017 | 0.8334 $\pm$ 0.0024 (-0.00378) | 0.8372 $\pm$ 0.0017 (+0.00378)         |
| Image missing    | 0.7762 $\pm$ 0.0059 | 0.7933 $\pm$ 0.0047 (+0.01702) | 0.7998 $\pm$ 0.0032 (+0.00653)         |
| Text missing     | 0.7658 $\pm$ 0.0094 | 0.7777 $\pm$ 0.0042 (+0.01181) | 0.7835 $\pm$ 0.0024 (+0.00584)         |
| Image noise 0.10 | 0.8366 $\pm$ 0.0016 | 0.8333 $\pm$ 0.0020 (-0.00329) | 0.8366 $\pm$ 0.0016 (+0.00329)         |
| Image noise 0.20 | 0.8359 $\pm$ 0.0017 | 0.8334 $\pm$ 0.0019 (-0.00253) | 0.8359 $\pm$ 0.0017 (+0.00253)         |
| Text noise 0.10  | 0.8360 $\pm$ 0.0018 | 0.8325 $\pm$ 0.0020 (-0.00341) | 0.8360 $\pm$ 0.0018 (+0.00341)         |
| Text noise 0.20  | 0.8334 $\pm$ 0.0019 | 0.8303 $\pm$ 0.0020 (-0.00304) | 0.8334 $\pm$ 0.0019 (+0.00304)         |

Table 4A. Classwise complete-missingness diagnostics (mean over seeds 42-51).

| Condition     | Model | Macro-F1 | F1 non-sarcastic | F1 sarcastic | Predicted sarcastic |
|---------------|-------|----------|------------------|--------------|---------------------|
| Image missing | B4    | 0.77625  | 0.78365          | 0.76885      | 53.59%              |
| Image missing | R1    | 0.79327  | 0.80956          | 0.77697      | 49.05%              |
| Text missing  | B4    | 0.76585  | 0.81203          | 0.71967      | 37.32%              |
| Text missing  | R1    | 0.77766  | 0.79757          | 0.75775      | 47.98%              |

## 4.5 Descriptive Uncertainty and Supporting Contrasts

Table 5 reports paired t intervals over ten seeds on the fixed test split. Table 6 and Figure 3 report the two-way analysis, which also resamples test items. The two-way ranges assess additional item sensitivity.

Table 5. Fixed-test descriptive paired macro-F1 contrasts over seeds 42-51 (\*B4–B3 is capacity-confounded).

| Comparison                   | Mean macro-F1 delta | 95% paired CI        | Positive seeds |
|------------------------------|---------------------|----------------------|----------------|
| Base: B1–B4                  | -0.03742            | [-0.04004, -0.03479] | 0/10           |
| Base: B2–B4                  | -0.05372            | [-0.05593, -0.05150] | 0/10           |
| Base: B4–B3*                 | +0.00259            | [0.00027, 0.00491]   | 8/10           |
| Base: B5–B4                  | -0.00085            | [-0.00304, 0.00134]  | 3/10           |
| RQ1: D1–B4                   | +0.00477            | [0.00123, 0.00832]   | 9/10           |
| RQ1: D1–B5                   | +0.00562            | [0.00271, 0.00853]   | 9/10           |
| RQ1 support: D4–B4           | +0.00331            | [-0.00143, 0.00805]  | 8/10           |
| RQ1 support: D4–B5           | +0.00416            | [-0.00003, 0.00835]  | 8/10           |
| RQ1 support: D1–D4           | +0.00146            | [-0.00151, 0.00444]  | 7/10           |
| RQ1 support: D1–D2           | +0.00189            | [-0.00097, 0.00476]  | 4/10           |
| RQ1 support: D1–D3           | +0.00032            | [-0.00255, 0.00318]  | 4/10           |
| RQ2: K3–K1                   | -0.00270            | [-0.00412, -0.00127] | 1/10           |
| RQ2: K3–K2                   | +0.00123            | [-0.00160, 0.00407]  | 7/10           |
| RQ2 support: K1–K0           | +0.00120            | [-0.00045, 0.00286]  | 5/10           |
| RQ3: R1–B4, clean            | -0.00378            | [-0.00592, -0.00165] | 0/10           |
| RQ3: R1–B4, image missing    | +0.01702            | [0.01080, 0.02325]   | 10/10          |
| RQ3: R1–B4, text missing     | +0.01181            | [0.00483, 0.01880]   | 10/10          |
| RQ3: R1–B4, image noise 0.10 | -0.00329            | [-0.00513, -0.00146] | 2/10           |
| RQ3: R1–B4, image noise 0.20 | -0.00253            | [-0.00467, -0.00039] | 3/10           |
| RQ3: R1–B4, text noise 0.10  | -0.00341            | [-0.00567, -0.00115] | 2/10           |
| RQ3: R1–B4, text noise 0.20  | -0.00304            | [-0.00526, -0.00082] | 1/10           |

Table 6. Selected paired two-way seed–item resampling sensitivity ranges (5,000 descriptive replicates).

| Contrast               | Observed delta | Seed-level 95% interval | Two-way 95% sensitivity range |
|------------------------|----------------|-------------------------|-------------------------------|
| D1–B4                  | +0.00477       | [0.00123, 0.00832]      | [-0.00157, 0.01114]           |
| K3–K1                  | -0.00270       | [-0.00412, -0.00127]    | [-0.00729, 0.00171]           |
| K3–K2                  | +0.00123       | [-0.00160, 0.00407]     | [-0.00372, 0.00625]           |
| R1–B4 clean            | -0.00378       | [-0.00592, -0.00165]    | [-0.00834, 0.00082]           |
| R1–B4 image missing    | +0.01702       | [0.01080, 0.02325]      | [0.00873, 0.02582]            |
| R1–B4 text missing     | +0.01181       | [0.00483, 0.01880]      | [0.00038, 0.02413]            |
| Route–R1 image missing | +0.00653       | [0.00220, 0.01086]      | [-0.00202, 0.01481]           |
| Route–R1 text missing  | +0.00584       | [0.00231, 0.00938]      | [-0.00222, 0.01388]           |
| Late fusion–B4         | -0.00417       | [-0.00621, -0.00213]    | [-0.01193, 0.00355]           |

### Paired two-way seed–item resampling sensitivity ranges

5,000 descriptive percentile replicates; positive values favor the first method.

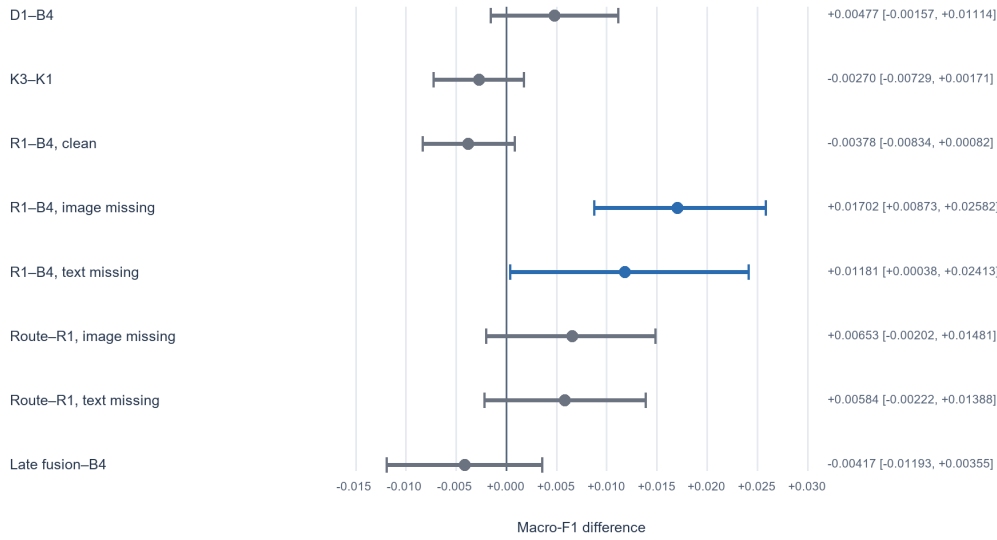


Figure 3. Selected paired two-way seed–item resampling sensitivity ranges; positive values favor the first method.

## 4.6 Exact-Text Overlap Sensitivity

Through a preliminary review, no same image were found, but 83 same text samples were found: 54 same label only samples, 17 opposite label only samples and 12 mixed label samples. We also found that the accuracy for same label only samples were high, more than 80%, and the accuracy for opposite label samples were low, only about 20%. This means that the model dose use duplicate texts as shortcuts. These repetitive samples were then deleted from the test, and recalculate the core values. This check does not address paraphrases, visual duplicates, or other artifacts.

Table 6A. Selected D, K, and R contrasts before and after excluding all 83 exact train-text repetitions.

| Contrast            | Full delta | Full 95% interval    | Unseen delta | Unseen 95% interval  | Positive unseen seeds |
|---------------------|------------|----------------------|--------------|----------------------|-----------------------|
| D1–B4 clean         | +0.00477   | [0.00123, 0.00832]   | +0.00432     | [0.00068, 0.00795]   | 9/10                  |
| K3–K1 clean         | -0.00270   | [-0.00412, -0.00127] | -0.00264     | [-0.00399, -0.00129] | 1/10                  |
| K3–K2 clean         | +0.00123   | [-0.00160, 0.00407]  | +0.00097     | [-0.00183, 0.00377]  | 8/10                  |
| R1–B4 clean         | -0.00378   | [-0.00592, -0.00165] | -0.00449     | [-0.00696, -0.00202] | 1/10                  |
| R1–B4 image missing | +0.01702   | [0.01080, 0.02325]   | +0.01734     | [0.01103, 0.02365]   | 10/10                 |
| R1–B4 text missing  | +0.01181   | [0.00483, 0.01880]   | +0.01423     | [0.00672, 0.02174]   | 10/10                 |

## 5. Discussion

### 5.1 What the Controlled Comparisons Show

The three methods all produced different outcomes. The shared and private decomposition method dose improved the model’s performance, since D1 has a better score than B4, but existing experiments couldn’t tell where exactaly the improvements came from. The generated knowledge method shows that not all information can improve the model’s performance, only when the generated knowledge can really help the model to decide can it improve the model’s performance, since K3 ks significantly worse than K1. Robust training is more like a trade-off, rather than a whole improvement.

## 5.2 Contribution

The methodological contribution is a workflow that assigns distinct controls to architecture, auxiliary information, training randomness, test-item sensitivity, and deployment alternatives.

## 6. Limitations

There are still limitations in the findings. All the experiments are using MMSD2.0 data, a fixed OpenCLIP ViT-B/32 encoder, 10 seeds and onesimplified implementation for each family. Thus, the results can only be constrained within this specific condition, and can't be directly extended to other different conditions. Another problem is the incomplete control of variables. Although B5 has the same parameters as D1, there network topology, parameter distribution and so on, are different. This is also the reason why it's unable to percisely identify why the shared and private decomposition method can improve the performance of the model. The robust training only covered when there are completely lost modal or simple gaussian noise, so it can't be generalized to other normal conditions when there are low quality inputs.

## 7. Conclusion

Through this experiment, we can find that shared and private decomposition can bring small improvements to the model, but the reason is still not clear; genereated knowledge can't bring improvements when the context is not selected to only include useful text; robust training can bring improvements when there are lost modal but also may bring reductions for all perfect inputs. Results shows that structure improvements, external knowledge and robustness all requires strict controlled experiments, to find out their real value for improving the performance.

## Appendix A. Parameter and Path Accounting

Table A1. Nominal parameters and structural participation in the tested formal implementations.

| Configuration | Total trainable | Direct inference graph | Auxiliary-objective path | Structural content path | Interpretive boundary                                                               |
|---------------|-----------------|------------------------|--------------------------|-------------------------|-------------------------------------------------------------------------------------|
| <b>B4</b>     | 525,570         | 525,570                | 0                        | 525,570                 | All trainable parameters lie on the direct inference graph.                         |
| <b>B5</b>     | 656,962         | 656,962                | 0                        | 656,962                 | Matches D1 total parameters, not architecture or allocation.                        |
| <b>D1/D2</b>  | 658,818         | 395,650                | 263,168                  | 658,818                 | Decoders receive reconstruction-objective gradients but do not set logits directly. |
| <b>D3/D4</b>  | 658,818         | 395,650                | 263,168                  | 395,650                 | Decoders receive no reconstruction-objective gradient.                              |
| <b>K0</b>     | 525,826         | 262,914                | 0                        | 262,914                 | Zero context and mask leave only base and classifier active.                        |
| <b>K1</b>     | 525,826         | 394,754                | 0                        | 394,754                 | Biases and gate can learn an availability-conditioned residual.                     |
| <b>K2/K3</b>  | 525,826         | 525,826                | 0                        | 525,826                 | Nonzero context makes the full nominal content path available.                      |

Direct-inference-graph counts describe structural participation, not effective predictive capacity. Auxiliary objectives can alter the learned encoders, and K path counts are structural upper bounds.

### A.1 Availability-Aware Route Resources

The three route heads contain 790,278 parameters in total, and 525,570 for R1. Under seed 42, their files occupy approximately 3.02 MiB versus 2.01 MiB. Approximate head MACs are 0.525M for B4 or R1 and 0.132M for a B1/B2 route decision. End-to-end latency was not retained.

## Appendix B. Supplementary Sensitivity and Examples

### B.1 Limited Equal-Budget Hyperparameter Sensitivity

The hyperparameter has been adjusted for each experiment: B4 with learning rate 0.0003 and dropout 0.2, D1 with learning rate 0.001 and dropout 0.2, K3 with learning rate 0.0003 and dropout 0.2 and R1 with learning rate 0.0003 and dropout 0.2 after the adjustment, there isn't a clear difference compared to the score without the adjustment.

## References

- [1] D. Wu, D. Yang, H. Shen, C. Ma, and Y. Zhou, "Resolving Sentiment Discrepancy for Multimodal Sentiment Detection via Semantics Completion and Decomposition," *Pattern Recognition*, vol. 172, article 112719, 2026. <https://doi.org/10.1016/j.patcog.2025.112719>
- [2] W. Dai, X. Li, Z. Wang, P. Hu, J. Qi, J. Peng, and Y. Zhou, "MIND: Improving Multimodal Sentiment Analysis via Multimodal Information Disentanglement," *arXiv:2401.11818*, 2024. <https://arxiv.org/abs/2401.11818>
- [3] W. Wang, L. Ding, L. Shen, Y. Luo, H. Hu, and D. Tao, "WisdoM: Improving Multimodal Sentiment Analysis by Fusing Contextual World Knowledge," in *Proceedings of ACM Multimedia 2024*, 2024. <https://doi.org/10.1145/3664647.3681403>
- [4] S. Zhang, J. Lian, G. Yu, B. Xu, and X. Ao, "GDCNet: Generative Discrepancy Comparison Network for Multimodal Sarcasm Detection," in *ICASSP 2026 – 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 13192–13196, 2026. <https://doi.org/10.1109/ICASSP55912.2026.11461651>
- [5] D. Wu, D. Yang, Y. Zhou, and C. Ma, "Robust Multimodal Sentiment Analysis of Image-Text Pairs by Distribution-Based Feature Recovery and Fusion," in *Proceedings of ACM Multimedia 2024*, pp. 5780–5789, 2024. <https://doi.org/10.1145/3664647.3680653>
- [6] Y. Wang, Q. Jia, C. Xu, F. Chen, Y. Liu, H. Zhang, L. Jin, L. Lu, and Z. Wang, "Conditional Information Bottleneck for Multimodal Fusion: Overcoming Shortcut Learning in Sarcasm Detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 31, pp. 26652–26660, 2026. <https://ojs.aaai.org/index.php/AAAI/article/view/39874>
- [7] D. Guo, C. Cao, F. Yuan, Y. Liu, G. Zeng, X. Yu, H. Peng, and P. S. Yu, "Multi-View Incongruity Learning for Multimodal Sarcasm Detection," in *Proceedings of COLING 2025*, pp. 1754–1766, 2025. <https://aclanthology.org/2025.coling-main.119/>
- [8] A. A. Ramakrishnan, A. A. Ramakrishnan, and D. Lee, "IRONIC: Coherence-Aware Reasoning Chains for Multi-Modal Sarcasm Detection," *arXiv:2505.16258*, 2025. <https://arxiv.org/abs/2505.16258>
- [9] J. Zhou, Y. Wu, Y. Zhang, Y. Zhang, Y. Liu, B. Huang, and C. Yuan, "SemIRNet: A Semantic Irony Recognition Network for Multimodal Sarcasm Detection," in *2025 10th International Conference on Information and Network Technologies (ICINT)*, pp. 158–162, 2025. <https://arxiv.org/abs/2506.14791>
- [10] X. Wang, Y. Zhang, and L. Jing, "Can Large Vision-Language Models Understand Multimodal Sarcasm?" in *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 5340–5345, 2025. <https://doi.org/10.1145/3746252.3760864>
- [11] J. Chen, H. Yu, S. Huang, S. Liu, and L. Zhang, "InterCLIP-MEP: Interactive CLIP and Memory-Enhanced Predictor for Multi-Modal Sarcasm Detection," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 22, no. 2, pp. 1–23, 2026. <https://doi.org/10.1145/3776561>
- [12] L. Qin, S. Huang, Q. Chen, C. Cai, Y. Zhang, B. Liang, W. Che, and R. Xu, "MMSD2.0: Towards a Reliable Multi-modal Sarcasm Detection System," *Findings of ACL 2023*, pp. 10834–10845, 2023. <https://doi.org/10.18653/v1/2023.findings-acl.689>
- [13] H. Yang, Y. Zhao, Y. Wu, S. Wang, T. Zheng, H. Zhang, Z. Ma, W. Che, and B. Qin, "Large Language Models Meet Text-Centric Multimodal Sentiment Analysis: A Survey," *arXiv:2406.08068*, 2024. <https://arxiv.org/abs/2406.08068>
- [14] S. Farabi, T. Ranasinghe, D. Kanojia, Y. Kong, and M. Zampieri, "A Survey of Multimodal Sarcasm Detection," in *Proceedings of IJCAI 2024*, pp. 8020–8028, 2024. <https://doi.org/10.24963/ijcai.2024/887>
- [15] S. Jana, A. Kundu, and S. R. Singh, "Think Twice Before You Judge: Mixture of Dual Reasoning Experts for Multimodal Sarcasm Detection," *arXiv:2507.04458*, 2025. <https://arxiv.org/abs/2507.04458>
- [16] Y. Cai, H. Cai, and X. Wan, "Multi-Modal Sarcasm Detection in Twitter with Hierarchical Fusion Model," in *Proceedings of ACL 2019*, pp. 2506–2515, 2019. <https://doi.org/10.18653/v1/P19-1239>
- [17] A. Radford et al., "Learning Transferable Visual Models From Natural Language Supervision," in *Proceedings of ICML 2021*, PMLR 139, pp. 8748–8763, 2021. <https://proceedings.mlr.press/v139/radford21a.html>
- [18] Q. McNemar, "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947. <https://doi.org/10.1007/BF02295996>
- [19] N. Xu, Z. Zeng, and W. Mao, "Reasoning with Multimodal Sarcastic Tweets via Modeling Cross-Modality Contrast and Semantic Association," in *Proceedings of ACL 2020*, pp. 3777–3786, 2020. <https://doi.org/10.18653/v1/2020.acl-main.349>
- [20] Z. Zhu, X. Cheng, G. Hu, Y. Li, Z. Huang, and Y. Zou, "Towards Multi-modal Sarcasm Detection via Disentangled Multi-grained Multi-modal Distilling," in *Proceedings of LREC-COLING 2024*, pp. 16581–16591, 2024. <https://aclanthology.org/2024.lrec-main.1441/>
- [21] B. Tang, B. Lin, H. Yan, and S. Li, "Leveraging Generative Large Language Models with Visual Instruction and Demonstration Retrieval for Multimodal Sarcasm Detection," in *Proceedings of NAACL 2024*, pp. 1732–1742, 2024. <https://doi.org/10.18653/v1/2024.naacl-long.97>
- [22] K. Li, Y. Chen, Q. Wu, W. Mai, F. Li, and Y. Xue, "Ambiguity-aware Multi-level Incongruity Fusion Network for Multi-Modal Sarcasm Detection," in *Proceedings of COLING 2025*, pp. 380–391, 2025. <https://aclanthology.org/2025.coling-main.26/>
- [23] J. Zhang, C. L. P. Chen, S. Li, and T. Zhang, "Incongruity-aware Tension Field Network for Multi-modal Sarcasm Detection," in *Proceedings of ACL 2025*, pp. 14499–14508, 2025. <https://doi.org/10.18653/v1/2025.acl-long.705>
- [24] H. Zhao, Y. Kong, Y. Xu, G. Gou, H. Xu, Y. Wang, and H. Zhang, "MMSD3.0: A Multi-Image Benchmark for Real-World Multimodal Sarcasm Detection," in *Proceedings of CVPR 2026*, pp. 37885–37895, 2026. [https://openaccess.thecvf.com/content/CVPR2026/html/Zhao\\_MMSD3.0\\_A\\_Multi-Image\\_Benchmark\\_for\\_Real-World\\_Multimodal\\_Sarcasm\\_Detection\\_CVPR\\_2026\\_paper.html](https://openaccess.thecvf.com/content/CVPR2026/html/Zhao_MMSD3.0_A_Multi-Image_Benchmark_for_Real-World_Multimodal_Sarcasm_Detection_CVPR_2026_paper.html)
- [25] mlfoundations, "OpenCLIP: An Open Source Implementation of CLIP," software repository, open\_clip\_torch version 3.3.0 used in this study, accessed July 2026. [https://github.com/mlfoundations/open\\_clip](https://github.com/mlfoundations/open_clip)
- [26] Zhipu AI, "GLM-4.1V-Thinking," official model and API documentation, accessed July 2026. <https://docs.bigmodel.cn/cn/guide/models/vlm/glm-4.1v-thinking>
- [27] E. B. Wilson, "Probable Inference, the Law of Succession, and Statistical Inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927. <https://doi.org/10.1080/01621459.1927.10502953>
- [28] B. Efron, "Bootstrap Methods: Another Look at the Jackknife," *The Annals of Statistics*, vol. 7, no. 1, pp. 1–26, 1979. <https://doi.org/10.1214/aos/1176344552>
- [29] S. Holm, "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979. <https://doi.org/10.2307/4615733>