DETECTING IMPLICIT SARCASM IN SPEECH BY CROSS-MODAL REASONING WITH MULTI-SOURCE KNOWLEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper focuses on detecting complex sarcasm emotions in speech. Simple emotions like happiness can be directly judged based on acoustic signals in speech. By contrast, sarcasm involves a conflict between surface features and underlying intention, e.g., using positive words to convey sadness. That is hard to detect by the sound or words alone. Moreover, implicit sarcasm often involves external knowledge related to commonsense, plots, scenes, etc., which is not presented in the given speech, resulting in insufficient clues for inferring the conflict. To solve these issues, we propose a new reasonable framework. We first encode the text and acoustic features in speech, and grasp their subtle correlations. That helps to find potential sarcastic conflicts in the heterogeneous space. We then retrieve the missing textual and prosodic knowledge related to the speech. They are necessary clues for inferring conflicts and are used to build an emotional graph. Based on it, we deduce a conflict chain between the literal words, true intention, and knowledge constraints by reinforced multi-hop inference. This chain can help to identify implicit sarcasm more accurately and explainably. Moreover, we create a large-scale dataset called *AEbSD* to conduct extensive evaluations. The results show our effectiveness and reliability.

1 INTRODUCTION

With the rapid development of interactive AI, grasping complex emotions in speech has become increasingly important, e.g., helping to understand the user's feelings in AI dialog, enabling better responses. Among emotions, sarcasm is widely used and challenging. It conveys attitudes and viewpoints in a unique way, i.e., the literal words and the inner intention are in conflict. Detecting such emotion has great value in applications like empathetic dialogue systems, educational assistants, and multimedia content analysis. For example, it can assist the educational AI to recognize the sarcastic utterance in a story and provide appropriate explanations for young listeners, preventing misinterpretation of the true attitude. Simple emotions can often be perceived by surface keywords and tone patterns. In contrast, sarcasm involves deliberate emotional disguise. As shown in Fig.(1), the speaker appears to enjoy the event, but his inner feeling is actually angry. There are no negative words. The speech tone is mild and steady, without obvious passive connotations. Relying merely on the surface words or tone would easily lead to misjudgment. Moreover, the implicit sarcasm is more subtle, even involves an in-depth reasoning about external common sense. That is, it is hard to identify the sarcasm merely by relying on the speech itself. We need to use some extra well-known facts like "*Weekend is for rest. It's annoying to work overtime on weekends when personal time is being occupied*" to make a correct inference. It is not trivial to infer hidden sarcastic intentions from the speech. This task is still less studied. Thus, we focus on it to fill the research gap.

To detect sarcasm, early work mostly focused on the text module using rule-based or machine learning methods, relying on features such as keywords, punctuation, and syntactic patterns. However, our task involves more modalities with subtle voice. The rules are hand-crafted, with poor scalability. Moreover, the literal features often contradict the inner sarcastic intention. Relying on them to predict will cause errors easily. Some studies turned to neural networks, encoding features in more modalities to better distinguish sarcasm. But text and voice are heterogeneous. Simple concatenation or weight fusion fails to grasp their correlations, making it hard to detect sarcastic conflicts. Some improved

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.

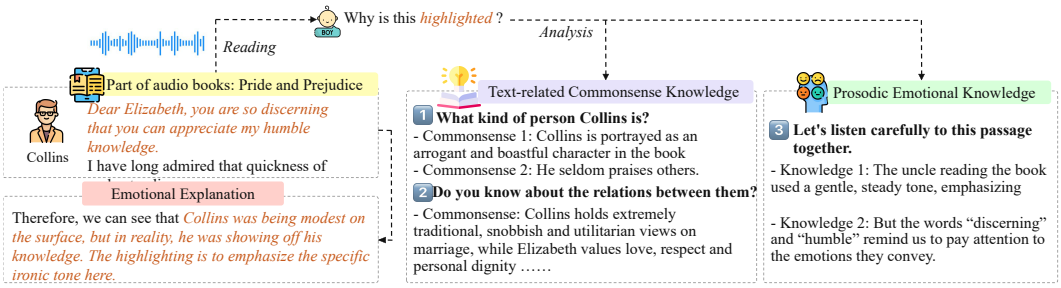


Figure 1: An example of implicit sarcasm speech.

methods, like mapping text and voice to a multimodal space, would cause ambiguity, e.g., the acoustic features of *increased speaking speed* would be wrongly associated with *positive words* instead of the sarcastic intention based on simple co-occurrence. Moreover, implicit sarcasm often involves background knowledge beyond the speech itself, like commonsense, prosody, etc. They do not appear in the speech, but are indispensable clues to make correct judgments. Even some methods try to incorporate knowledge into the model, but they only do so as additional features, rather than as the basis to deduce a logical reasoning chain to a correct result. Moreover, the interpretability of this black box model is also poor.

Motivated by above observations, we propose a knowledge-augmented framework to deduce implicit sarcasm in speech in an explainable way. To discover the hidden sarcastic conflicts, we first encode text and acoustic features in speech and capture their subtly correlated clues, such as a certain word paired with an abnormal tone, etc. This can utilize the complementary contexts of multiple modalities to enhance the model's robustness. We then retrieve knowledge related to the text and voice, such as plot background, commonsense, emotional prosody, etc. Instead of embedding them as auxiliary features into the model, we use them as the logical constraint for the reasoning process of sarcastic conflict. That can supplement the missing but salient clues to better infer sarcasm. Considering implicit sarcasm often involves multi-hop conflicts, traditional methods roughly feed all content into the classifier. This black-box method has poor interpretability and is hard to infer the multi-step indirect conflicts. We thus build an emotion graph, converting the abstract scattered contradictions among surface words, phonetic rhythm, and various external knowledge into calculable topological relations. Based on this graph, we derive a sarcastic conflict chain via reinforced multi-hop reasoning. This chain can bridge the gap between observable speech signals and hidden sarcastic intentions. It can also well explain why it is judged as sarcasm, enhancing the model's interpretability. To support evaluation on this task, we build a large-scale dataset named *AEbSD*. We conduct extensive experiments on it and several classic datasets. The results show the effectiveness of our method. Even when there are no explicit satirical signals in speech, we can still find implicit sarcasm.

The main contributions of this paper include,

- We propose a valuable task to detect implicit sarcasm in speech. Unlike simple emotions that can be judged by surface feature matching, our task is more challenging, involves hidden emotional conflicts and cross-model reasoning with external knowledge. We reveal limitations of existing methods on this task.
- We propose a new method with good interpretability. It builds an emotion graph with multi-source knowledge to better capture potential correlations and subtle conflicts, helping infer implicit sarcasm more accurately.
- We construct a large-scale evaluation dataset for the task to facilitate research in this field. Extensive experiments are conducted to examine the effectiveness of our approach.

2 METHODOLOGY

As shown in Fig.(2), given a speech signal a and its text content t, we aim to identify whether they convey implicit sarcasm emotion via three steps. We first analyze the inputs to capture subtle and complementary linguistic and acoustic clues. We then retrieve relevant knowledge to supplement the

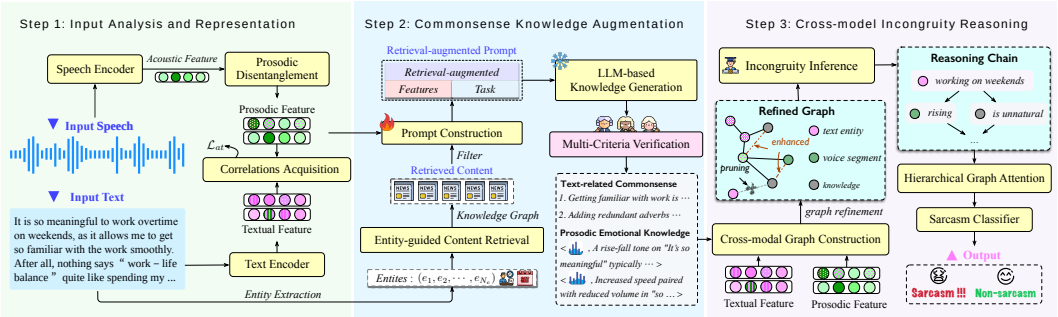


Figure 2: Pipeline of our implicit sarcasm reasoning framework.

missing but essential clues for reasoning. Based on them, we build an emotion graph to help infer the emotional conflict chain for implicit sarcasm detection. Next, we elaborate on each part.

2.1 Inputs Analysis and Representation

We first encode the input t as $\boldsymbol{E}_t$ by *BERT* to capture the sarcastic context. Considering there would be irrelevant and redundant terms in t, we then purify it by suppressing these terms and amplifying sarcasm-sensitive ones, as $\boldsymbol{H}_t = \mathcal{F}^t_{\text{ref}}(\boldsymbol{d}_t, \boldsymbol{E}_t)$, where $\mathcal{F}^t_{\text{ref}}$ is a feed-forward network with cross-attention, $\boldsymbol{d}_t$ is a learnable denoising query. Similarly, we encode acoustic features $\boldsymbol{E}_a$ of the voice a via *WavLM*. Since these basic features (such as speed, duration, energy, etc.) do not carry text semantics, there will be weak discrimination. For example, the same slow and elongated pronunciation could be interpreted as fatigue, sadness, or sarcasm depending on the text. We thus use a text query $\boldsymbol{d}_t$ as a strong guidance to regularize the interpretation of acoustic features, as $\boldsymbol{F}_a = \mathcal{F}^a_{\text{ref}}(\boldsymbol{d}_a, \boldsymbol{d}_t, \boldsymbol{E}_a)$. Besides, such features may contain irrelevant variations caused by background noise, recording conditions, and speaker differences. We design a dual-branch disentanglement module to decompose $\boldsymbol{F}_a$ into sarcasm-related $\boldsymbol{H}_a$ and irrelevant $\boldsymbol{H}^{\text{irr}}_a$. The module is optimized by Eq.(1), where $r(\cdot)$ is a reconstruction function, β balances reconstruction and orthogonality constraints, $|\cdot|^2_2$ is the squared *L2* norm, $|\cdot|^2_F$ denotes the squared *Frobenius* norm.

$$\mathcal{L}_{\text{dis}} = \left|\boldsymbol{F}_a - r(\boldsymbol{H}_a, \boldsymbol{H}^{\text{irr}}_a)\right|^2_2 + \beta \left|\boldsymbol{H}^\top_a \boldsymbol{H}^{\text{irr}}_a\right|^2_F . \tag{1}$$

We observed that in sarcastic expressions, certain words (such as “*really great*”) are often deliberately emphasized, prolonged, or spoken in a lower tone. These subtle differences are imperceptible, but they are crucial for identifying disguised sarcasm. A global representation is hard to model these subtle but important clues. We thus segment $\boldsymbol{H}_a$ based on *CTC* timestamps, dividing it into equal-length segments corresponding to each word’s actual pronunciation start and end frames, as $\boldsymbol{H}_a = \{\boldsymbol{h}_{a1}, \boldsymbol{h}_{a2}, \cdots\}, \boldsymbol{h}_{an}$. Since sarcastic prosody may span across multiple adjacent words, strict segmentation may miss continuous variations. We employ locally constrained attention A_{ij}, as Eq.(2), where w controls the interaction range.

$$A_{ij} = \begin{cases} 0, & |i-j| \le w, \\ -\infty, & |i-j| > w. \end{cases} \tag{2}$$

To identify whether the text t and voice a contain emotional conflicts, we project them into a shared space and compute Eq.(3), where Inc_i denotes the inconsistency score of the i-th word, $p_t(i)$ and $p_a(i)$ represent textual and prosodic polarity, $\sigma(\cdot)$ is the sigmoid function, and κ controls sensitivity to polarity inconsistencies.

$$Inc_i = \frac{1}{2}|p_t(i) - p_a(i)|\sigma(-\kappa p_t(i) p_a(i)). \tag{3}$$

Since there are usually a few words that contain sarcastic evidence, averaging all word-level scores may weaken critical regions. We explore *log-sum-exp* pooling to derive a sentence-level inconsistency, as $\boldsymbol{C}_{at}[i,j] = Inc_i \left(\frac{1}{K}\sum_{k=1}^{K} \alpha^{(k)}_{ij}\right)$, where $|i-j| \le w$, $\boldsymbol{C}_{at}$ is a fine-grained conflict matrix, K is the number of attention heads, and $\alpha^{(k)}_{ij}$ denotes the attention weight between the i-th word and j-th

acoustic segment.

To encourage the model to focus on high-conflict regions, we employ contrastive learning, as Eq.(4), where ℓ_i denotes the word-level loss, $\mathrm{sg}(\cdot)$ prevents inconsistent factors from affecting feature optimization, and γ controls the weighting strength.

$$\mathcal{L}_{at}^{\mathrm{con}} = \frac{\sum_i (1 + \gamma\,\mathrm{sg}(Inc_i))\ell_i}{\sum_i (1 + \gamma\,\mathrm{sg}(Inc_i))}. \tag{4}$$

Considering normal emotional variations may also cause cross-modal differences, we regularize sentence-level inconsistency $Inc_{\mathrm{sent}}^{(i)}$ using the sarcasm labels, as Eq.(5), where y_i denotes a label, $\mathcal{B}$ is the batch size, η balances sarcastic samples, and m is the margin for enforcing sufficient conflict.

$$\mathcal{L}_{\mathrm{inc}} = \frac{1}{\mathcal{B}} \sum_i \Big[(1 - y_i) Inc_{\mathrm{sent}}^{(i)} + \eta y_i \max\left(0, m - Inc_{\mathrm{sent}}^{(i)}\right) \Big]. \tag{5}$$

By jointly optimizing all losses, we derive a matrix $\boldsymbol{C}_{at}$, which can better capture correlated conflict clues between text and voice at a fine-grained level.

2.2 Multi-source Knowledge Augmentation

Since the implicit sarcasm in speech often lacks ironic words and exaggerated prosody, the surface text/ voice features alone are hard to infer the inherent sarcastic intention. We propose to supplement the missing text-related commonsense C_t and prosodic emotional clues K_{pe}, using them to bridge the gap in the reasoning process of implicit sarcasm. We observe that knowledge graphs mostly consist of facts and lack emotion-related knowledge. Relying solely on ***LLMs*** is prone to yield general context-independent content. To address it, we employ a hybrid method. We first extract entities from the text t, and use them to retrieve relevant facts and emotional cues from multiple sources, such as *Wikipedia*, *ConceptNet*, *ATOMIC*, *COMET*, *ECoK*, and *SenticNet*, etc. Based on them, we then construct appropriate prompts to guide the ***LLMs*** to generate context-specific knowledge. To balance accuracy, redundancy, and coverage, we select the top-N retrieved content. To obtain prosodic emotional knowledge, our retrieval-augmented prompts consider both voice features and the retrieved text, and integrate them by Eq.(6), where each $\mathcal{M}$ denotes a projection matrix, ***MHA*** denotes multi-head attention, and H_t, H_a, and H_r denote the textual features, prosodic features, and the embedding of retrieved content, respectively.

$$\begin{aligned} H'_{ra} &= \mathrm{MHA}\left(H_t\mathcal{M}'_q; H_r\mathcal{M}'_k; H_r\mathcal{M}'_v\right), \\ H_{ra} &= \mathrm{MHA}(H'_{ra}\mathcal{M}_q; H_a\mathcal{M}'_k; H_a\mathcal{M}'_v). \end{aligned} \tag{6}$$

The final prompt input into ***LLMs*** consists of three parts, including the retrieval context H_{ra}; the text and prosodic features, H_t and H_a, which serve as the base to infer their intrinsic correlations; and the task descriptions, including the role, task goal, output format, conditions and constraints, etc. They are encoded as H_s by ***BERT***. Details of prompts are shown in Appendix A.1. We then concatenate them as $H'_p = [H_{ra} \| H_t \| H_a \| H_s]$. Since the dimensions of H'_p and ***LLMs*** are different, we utilize a linear projection layer to map H'_p into a space with the same dimension of ***LLMs***, as $\mathcal{M}_{lp}(H'_p)$. To improve training efficiency, we freeze the parameters of ***LLMs*** and let them yield C_t and K_{pe}.

Considering the output of ***LLMs*** may contain noise, we employ a multi-criteria filter (Li et al., 2025). It assesses the quality of retrieved content from three perspectives. (1) ***Relevance*** to the input text-voice, we compute their similarity as $f_{ks} = \delta \cdot \mathrm{Sim}(H_t, H_r) + (1 - \delta) \cdot \mathrm{Sim}(H_a, H_r)$, where $\delta \in (0, 1)$ is a coefficient to balance the speech signal-to-noise ratio. In weak intonation sarcastic scenarios, δ increased; otherwise, it decreased. We remove the irrelevant content with a score f_{ks} below a threshold $\tilde{\epsilon}$. (2) ***Rationality***: To derive the satirical conflict, the useful content often should be close to the negative emotional tone and stay away from the positive literal text. For example, when the speaker expresses sarcasm about working overtime in a calm tone in Fig.(1), the retrieved content "*Overtime work is a positive experience*" is noise and can be filtered out. Its score should be low. We thus compute it as $f_{re} = \sigma(\tau \cdot \Delta P);\ \Delta P = \|P_r - P_a\|_2 - \|P_r - P_t\|_2$, where σ is a *sigmoid* function and τ is the scaling coefficient, P_r, P_t and P_a are the polarity of the content, input text and voice, respectively. The polarity is predicted by an ***LLM***-based sentiment classifier (i.e., ***Qwen-7b***). (3) ***Adaptability***: The implicit sarcasm is mostly triggered by instantaneous intonation perturbations. The content should be able to match this variation. We calculate it as $f_{ad} = \frac{1}{T}\sum_{i=1}^{T} \mathrm{Sim}(H_r, H_a^i)$, where H_a^i is the temporal emotion feature of the i-th frame of speech.

2.3 CROSS-MODAL INCONGRUITY REASONING

Traditional methods often roughly read all content and inputs to learn mappings between features and sarcastic labels. The weak sarcastic signals that conflict with each other will be smoothed out, resulting in missed detections. In addition, it is hard to explain the emotional conflicts. We thus build an emotion graph to explicitly grasp potential relations. This facilitates long indirect reasoning, and helps to uncover subtle inconspicuous conflicts.

Graph Construction and Refinement. We view textual and commonsense entities, prosodic, and emotional knowledge features as nodes $\mathcal{V}$. The edge E of the graph $\mathcal{G} = \{\mathcal{V}, E\}$ is composed of four aspects: (1) We use a parser to capture the dependency relations between entities in the text t and retrieved C_t. For each relation, we establish an edge. (2) We add edges between the correlated text entity and voice fragment. (3) If there is a knowledge triple for the prosodic emotional clues K_{pe} and the voice fragment, we create an edge. (4) To capture long-tail relations, we calculate the cosine similarity between unconnected nodes, as $sim(H_i, H_j)$. We add an edge for node pairs with a similarity exceeding the threshold β.

To reduce redundancy and noise, we refine the graph. We first calculate the edge weight based on $\omega_{v,i} = \text{sigmoid}\big(\text{FFN}\big([h_v \| h_i \| \mathcal{A}[v,i]]\big)\big)$, where *FFN* denotes a feed-forward network, $\mathcal{A}$ is a matrix to represent the graph structure. We then select the top-l edges based on ω. To ensure the stability of the selection process, we utilize linear programming to regularize it, as $\text{argmax}_{\mathcal{S}_p \in \mathcal{C}} \langle \mathcal{S}_p \cdot \mathcal{M}_v^L \rangle$, where $\mathcal{S}_p$ is the one-hot sampling pointers, $\mathcal{M}_v^L$ is the weight matrix at L^{th} interactive computation. As shown in Eq.(7), $\mathcal{C}$ is used to guarantee the top-l one-hot property (i.e., each node only retains l optimal edges) and the rationality of score ranking. N_p denotes the number of neighbors. In this way, we can derive an optimal and informative graph.

$$\mathcal{C} = \begin{cases} \sum_{i=1}^{N_p} \mathcal{S}_p^{ij} = 1, & \forall j \leq L, \\ \mathcal{S}_p^{ij} \geq \mathcal{S}_p^{i(j+1)} \geq 0, & \forall i \leq N_p. \end{cases} \tag{7}$$

Due to the discreteness of the one-hot property and the discontinuity of the top-l operation, the sampling process is non-differentiable. To tackle it, we employ a *perturbation maximization* strategy (Berthet et al., 2020) for training, as Eq.(8), where ϑ controls the proportion of injected *Gaussian Noise* $\mathbb{U}$. It can convert the discrete top-l selection into a continuous optimizable process, facilitating gradient-based optimization of the model. Finally, we integrate the optimized edge weights to obtain a refined graph, represented as a matrix $\hat{\mathcal{A}}$.

$$\mathcal{S}_p^{\vartheta} = \mathbb{E}_{\mathbb{U}}[\text{argmax}_{\mathcal{S}_p \in \mathcal{C}} \langle \mathcal{S}_p, \mathcal{M}_v^L + \vartheta U \rangle]. \tag{8}$$

Conflict Inference. To enhance efficiency, we develop a reinforced reasoner. Based on a dedicated reward, it can avoid irrelevant paths in $\hat{\mathcal{G}}$ adaptively, and well explore the ones with sarcastic conflicts to form a reasoning chain. First, we initialize the policy network $\pi_\theta(a_z|s_z)$ by *LLMs* (i.e., *LlaMA3-8B*), where s_z represents the state (e.g., the current node, the graph structure), and a_z is the action to select the next-hop node from a candidate adjacency set. This $\pi_\theta(\cdot)$ has strong cross-modal semantic understanding capabilities and provides a good initialization. To reduce the computing cost, we adopted the *Low-Rank Adaptation* (*LoRA*) technique to freeze most of the parameters of *LLMs* and only train some adapted ones by the warm-up data. To reduce bias of single data, we design a diversified data sampling strategy to cover various reasoning scenarios. In each warm-up iteration, we randomly select a subgraph $\mathcal{G}_s^i$ and conduct trajectory sampling on it to obtain $\tau_z = \{(s_0, a_0, r_0), (s_1, a_1, r_1), \ldots, (s_z, a_z, r_z)\}$. To better learn the policy, we adopt a gradient-based warm-up objective, as presented in Eq.(9), where $\pi_\theta(a_z|s_z)$ denotes the policy probability of taking action a_z under state s_z, R_z is the discounted return, and b is a constant term. That can help to learn the initial interactive patterns, such as the priority jump relations of nodes. During reinforcement training, the model samples an action a_z based on the policy $a_z \sim \pi_\theta(a_z|s_z)$ and obtains a reward r_z for each step z. To facilitate exploring the best reasoning path, our reward adopts three metrics, including *node–query correlation*, *edge-level consistency*, and *cross-modal alignment*. Details of rewards are shown in Appendix A.3.

$$\mathcal{L}_{warm} = -\mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_z \log \pi_\theta(a_z|s_z) \cdot (R_z - b) \right]. \tag{9}$$

Further, we adopt *PPO* (Schulman et al., 2017) to optimize the policy network. *PPO* can enhance training stability and avoid performance fluctuations by regularizing the deviation between new and old policies, as $r_z(\theta) = \frac{\pi_\theta(a_z|s_z)}{\pi_{\theta_{\text{old}}}(a_z|s_z)}$. The loss is formulated as Eq.(10), where $\hat{A}_z$ is the advantage estimate and ϵ is a proximal constraint coefficient.

$$\mathcal{L}_{\text{clip}}(\theta) = -\mathbb{E}_z\left[\min\left(r_z(\theta)\hat{A}_z, \text{clip}\left(r_z(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_z\right)\right]. \tag{10}$$

To boost early exploration, we adopt an entropy regularization term, as $\mathcal{L}_{ent}(\theta) = -\beta_H\,\mathbb{E}_z[\mathcal{H}(\pi_\theta(\cdot|s_z))]$, where $\mathcal{H}$ denotes the policy entropy and β_H is a weighting coefficient. The final *PPO* objective integrates the policy, value, and entropy, as Eq.(11), where the value loss $\mathcal{L}_{vf}$ is denoted as $\mathcal{L}_{vf} = \frac{1}{2}\mathbb{E}_z\left[(V_\theta(s_z) - R_z)^2\right]$, c_b is the weight.

$$\mathcal{L}_{ppo}(\theta) = \mathcal{L}_{clip}(\theta) + c_b\mathcal{L}_{vf}(\theta) + \mathcal{L}_{ent}(\theta). \tag{11}$$

To accelerate gradient convergence, we further employ the *generalized advantage estimation*, as $\hat{A}_z = \sum_{l=0}^{z_{max}-z}(\gamma\lambda)^l\,\delta_{z+l}$, where $\delta_z = r_z + \gamma V_\theta\,(s_{z+1}) - V_\theta(s_z)$, λ controls the bias–variance trade-off. Upon convergence, the optimized policy π_θ^* is deployed to derive a final reasoning chain to reflect cross-modal sarcastic incongruities. Afterward, we utilize *hierarchical graph attention* (Wang et al., 2019) to encode this chain, as H_c. We then pass it through a single-layer *MLP* with a *sigmoid* activation to predict $\hat{y}$. The binary cross-entropy loss is defined as Eq.(12). The overall training loss is the weighted aggregation of $\mathcal{L}_{warm}$, $\mathcal{L}_{ppo}$, and $\mathcal{L}_{sd}$.

$$\mathcal{L}_{sd} = -(y\log(\hat{y}) + (1-y)\log(1-\hat{y})) \tag{12}$$

3 Evaluations

We conducted experiments with qualitative and quantitative analysis to evaluate our approach.

3.1 Dataset Construction and Settings

Early datasets, such as *Tweet* (Zhang et al., 2016) and *Ghosh* (Ghosh & Veale, 2017), mainly supported text-based sarcasm detection, but did not contain voice data. Current multimodal datasets, such as *MMSD2.0* (Qin et al., 2023), relied on the contrast between visual clues and text to indicate sarcasm. *MUStARD* (Castro et al., 2019) targeted on the video-related sarcasm based on the *TV* show clips. Their sarcastic expressions depended on scene pictures, actions, etc. This would lead to misjudgments when directly transferred into the speech domain. Moreover, they mostly focused on simple emotion instead of the implicit sarcasm that involved complex cross-modal and hidden conflicts. We thus built a large-scale sarcasm dataset, named *AEbSD*. It comprised 10834 voice-text pairs, with a relatively balanced class distribution. We partitioned it into *training, validation, and testing* sets with a 7:2:1 ratio. It was built on sources of audiobooks that contain high-quality recordings with rich emotional expressions. We first collected audiobooks from *LibriVox* and their transcripts from *Project Gutenberg*. They are copyright-free public repositories with legal and ethical standards. We applied the tool *WhisperX* to yield sentence-level voice-text pairs based on timestamps. To efficiently form sarcastic cases, we first designed a customized prompt based on three aspects, including *prosodic clarity*, *signal-to-noise ratio*, and *contextual coherence*. We used it to guide the *LLM* to score each text-voice candidate pair. Considering implicit sarcasm involved reasoning over external knowledge, we manually reviewed them to check whether they had cross-modal incongruity and external knowledge dependency. Samples that satisfied all criterion were retained as positive instances; otherwise, negative. We invited fifteen annotators to label each candidate. Each was cross-validated by three extra volunteers. Majority voting was applied to resolve disagreements. Construction details were shown in Appendix B.

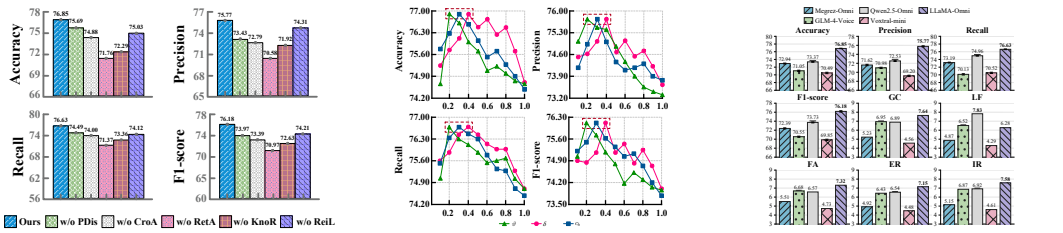
We adopted 4 evaluation metrics that were popular on the sarcasm task, including *Accuracy(Acc)*, *Recall(Rec)*, *Precision(Pre)*, and *F1 score*. More details were provided in Appendix C.1.

3.2 Model Comparison

We compared our method against nine popular baselines on five categories: (1) prompt-based methods, including *SpeechPrompt v2* (Chang et al., 2023) and *LTU-AS* (Gong et al., 2023), using prompt-tuning

Table 1: Performance comparison, where **bold** and underlined denote the best and second-best results. All results are significant with a paired t-test ($p < 0.05$).

Method	Acc	Pre	Rec	F1
SpeechPrompt v2	65.95±0.13	66.16±0.11	65.18±0.09	65.67±0.07
LTU-AS	67.67±0.15	67.47±0.12	68.11±0.14	67.79±0.16
MUStARD++	62.55±0.08	62.33±0.17	63.26±0.10	62.79±0.19
PS2RI	62.13±0.12	62.24±0.06	61.49±0.15	61.86±0.05
Sync-TVA	66.72±0.18	66.96±0.14	65.88±0.07	66.42±0.11
GLoMo	67.14±0.09	67.37±0.16	66.35±0.13	66.86±0.08
DPDF-LQ	67.79±0.16	67.57±0.10	68.29±0.18	67.93±0.14
Qwen-Audio	69.92±0.11	70.13±0.19	69.29±0.12	69.71±0.06
GPT-4o	69.30±0.17	69.42±0.08	68.89±0.15	69.15±0.13
Ours	**76.09**±0.06	**75.77**±0.12	**76.63**±0.09	**76.20**±0.15



(a) Ablation studies.

(b) Sensitivity analysis of different settings.

(c) Comparison between LLM and human evaluations.

Figure 3: Additional analyses of model components, parameter sensitivity, and evaluation quality.

on *LLM* to detect voice; (2) multimodal emotion recognition methods, including *MUStARD++* (Ray et al., 2022) and *PS2RI* (Fang et al., 2024), using dual fusion of text and speech features for judgment. (3) graph-based methods, i.e., *Sync-TVA* (Deng et al., 2025), modeling the temporal relations between text and speech via graphs; (4) cross-modal methods, including *GLoMo* (Zhuang et al., 2024), and *DPDF-LQ* (Zhou et al., 2025), representing speech emotion via multi-level feature interaction; (5) general multimodal *LLM*, including *Qwen-Audio* (Chu et al., 2023) and *GPT-4o* (Achiam et al., 2023).

As shown in Table 1, our model achieved the best overall performance. Compared to general *LLM*, our *F1* score exceeded that of *GPT-4o* by nearly 7%. This indicated we can better detect complex emotions in speech by customized design and knowledge enhancement. General models are prone to being misled by surface terms when faced with implicit sarcasm with disguised features. Besides, our model significantly outperformed *MUStARD++* and *PS2RI* in terms of all metrics. That indicated capturing the subtle correlations between voice and text is helpful in identifying local imperceptible conflicts. Compared to *GLoMo* and *DPDF-LQ*, our model achieved notable improvements in terms of accuracy and *F1* score. That reflected the value of incorporating external knowledge for detection. Moreover, we surpassed *Sync-TVA* in terms of accuracy by 9.37%. That indicated the conflict chains we deduced based on the graph, rather than injecting the entire graph into the model, can better detect implicit sarcasm. Moreover, we further evaluated it on popular datasets, including *IEMOCAP* (Busso et al., 2008), *MELD* (Poria et al., 2019), and *MUStARD++*. As presented in Appendix 3.6, we observed consistent improvements, indicating a good generalizability of our model.

3.3 Ablation Studies

We performed ablation studies to verify the effectiveness of our key modules, including (1) *PDis* that removed the prosodic features disentanglement module; (2) *CroA* that disabled the token-level correlations acquisition; (3) *RetA* that disabled part of the *RAG* module and let *LLMs* to directly yield content based on the input; (4) *KnoR* that threw away the entire *RAG* module; (5) *ReiL* that replaced the reinforced reasoning module, inputting the entire graph into the model directly instead of the conflict chain. As shown in Fig.(3a), removing any module led to noticeable degradation. That indicated they were all indispensable for discerning implicit sarcasm. Moreover, we studied the individual contributions of different knowledge sources, with detailed results provided in Section 3.5.

Table 2: Performance on benchmarks of *IEMOCAP*, *MELD*, and *MUStARD++*. Results are reported as mean ± std over 5 independent runs. Best results are in **bold**; second-best are underlined. "—" means the method did not match that benchmark and thus was not evaluated on it.

Method	IEMOCAP		MELD	MUStARD++		
	WeiAcc	UnweiAcc	WeiF1	WeiPre	WeiRec	WeiF1
Wav2vec2-base	$54.38_{\pm 0.28}$	$55.42_{\pm 0.24}$	$31.94_{\pm 0.22}$	—	—	—
HuBERT-base	$62.73_{\pm 0.29}$	$62.91_{\pm 0.25}$	$37.72_{\pm 0.27}$	—	—	—
WavLM-base	$71.68_{\pm 0.22}$	$70.34_{\pm 0.26}$	$42.53_{\pm 0.29}$	—	—	—
Whisper-base	$62.84_{\pm 0.27}$	$61.03_{\pm 0.28}$	$34.31_{\pm 0.23}$	—	—	—
SpeechFormer	$62.93_{\pm 0.25}$	$64.52_{\pm 0.22}$	$41.87_{\pm 0.28}$	—	—	—
DWFormer	$\underline{71.82}_{\pm 0.21}$	$\underline{72.84}_{\pm 0.19}$	$\underline{46.71}_{\pm 0.26}$	—	—	—
ENT	$71.61_{\pm 0.27}$	$72.43_{\pm 0.25}$	$46.28_{\pm 0.28}$	—	—	—
TENT	$71.44_{\pm 0.26}$	$71.83_{\pm 0.24}$	$45.13_{\pm 0.29}$	—	—	—
PredGaze	—	—	—	$71.06_{\pm 0.28}$	$71.06_{\pm 0.25}$	$71.06_{\pm 0.27}$
FIES	—	—	—	$71.48_{\pm 0.26}$	$71.43_{\pm 0.29}$	$71.45_{\pm 0.24}$
ABCA-IMI	—	—	—	$71.90_{\pm 0.29}$	$69.90_{\pm 0.27}$	$70.90_{\pm 0.28}$
SIB	—	—	—	$69.67_{\pm 0.27}$	$69.54_{\pm 0.25}$	$69.39_{\pm 0.26}$
DIB	—	—	—	$70.96_{\pm 0.25}$	$71.12_{\pm 0.27}$	$70.98_{\pm 0.24}$
MUStARD++	—	—	—	$71.02_{\pm 0.28}$	$70.80_{\pm 0.29}$	$70.98_{\pm 0.26}$
Speechprompt v2	—	—	—	$\underline{78.33}_{\pm 0.27}$	$58.06_{\pm 0.29}$	$73.47_{\pm 0.28}$
MCIB	—	—	—	$77.18_{\pm 0.26}$	$76.30_{\pm 0.28}$	$\underline{76.85}_{\pm 0.25}$
GPT-4o	—	—	—	$62.66_{\pm 0.29}$	$\underline{83.90}_{\pm 0.27}$	$71.74_{\pm 0.28}$
Ours	$\mathbf{72.83}_{\pm 0.18}$	$\mathbf{73.04}_{\pm 0.16}$	$\mathbf{47.12}_{\pm 0.22}$	$\mathbf{75.77}_{\pm 0.19}$	$\mathbf{85.36}_{\pm 0.17}$	$\mathbf{80.21}_{\pm 0.18}$

3.4 Evaluation of Parameter Sensitivity

We analyzed the key hyperparameters used in our model, including the relevance weight δ in Section 2.2, the noise weight ϑ in Eq.(8), and the coefficient c_b in Eq.(11). For each parameter, we tuned it with an interval of 0.1 over the range $(0, 1]$, while fixing the others at their optimal values from the previous univariate analysis. As shown in Fig.(3b), we observed that the best result was achieved when the relevance weight δ was 0.4, the noise weight ϑ was 0.2, and the coefficient c_b was 0.3. This indicated that properly balancing knowledge relevance, noise injection, and strategy-value optimization could improve performance. More details of hyperparameters are provided in Appendix C.1.

3.5 Evaluations on External Knowledge

To validate the impact of knowledge retrieval tools on our model, we studied different *LLMs*, including *Megrez-Omni*, *GLM-4-Voice*, *Qwen2.5-Omni*, *Voxtral-mini*, and *LlaMA-Omni*. As presented in Fig.(3c), our results exhibited minor fluctuations and were still better than most baselines. That reflected the robustness of our model with designs like filtering and refinement. To verify the influence of the knowledge quality on our model, we further conducted human evaluation in terms of the metrics of *Grammatical Correctness (GC)*, *Language Fluency (LF)*, *Factual Accuracy (FA)*, *Emotional Relevance (ER)*, and *Input Relevance (IR)*. As shown in Fig.(3c), we observed *LlaMA-Omni* can better understand text and voice. It can supplement useful knowledge clues to help infer implicit sarcasm. More details were shown in Appendix C.3.

3.6 Evaluations on Generalization Ability

Further, we evaluated the generalizability of our model on two extra popular benchmarks for emotion detection, i.e., *IEMOCAP* (Busso et al., 2008) and *MELD* (Poria et al., 2019), as well as the multimodal sarcasm benchmark *MUStARD++* (Ray et al., 2022). For *IEMOCAP* and *MELD*, we adopted *Weighted Accuracy* (*WeiAcc*) and *Unweighted Accuracy* (*UnweiAcc*) as evaluation metrics for *IEMOCAP*, and *Weighted F1-score* (*WeiF1*) for *MELD*, with baselines adopted from (Zhao et al., 2025). For *MUStARD++*, we reported the metrics of *Weighted Precision* (*WeiPre*), *Weighted Recall* (*WeiRec*), and *WeiF1*, with baselines in (Wang et al., 2025). As shown in Table 2, our model achieved the best performance on all benchmarks. On *IEMOCAP*, it surpassed the second-best *DWFormer* by 1.01% and 0.20% in terms of *WeiAcc* and *UnweiAcc*, respectively. On *MELD*, our model outperformed the second-best by 0.41% in terms of *WeiF1*. On *MUStARD++*, we achieved a *WeiF1* of 80.21%, surpassing the second-best (*MCIB*) by 3.36%. We also obtained a notably high *WeiRec* (85.36%), exceeding the strongest recall baseline (*GPT-4o*) by 1.46%. That reflected our prosodic disentanglement and conflict inference modules can well adapt various domains.

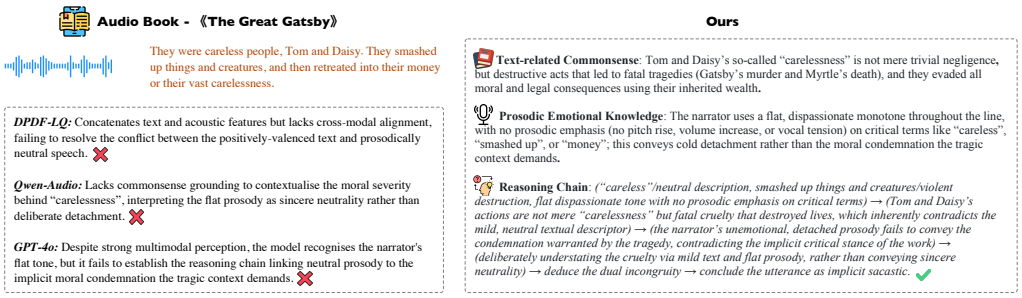


Figure 4: Case Study.

3.7 Case Studies

Moreover, we analyzed the output results and observed our model could accurately identify implicit sarcasm. As depicted in Fig.(4), other baselines with top-3 performance failed to uncover the key sarcastic conflicts in speech. In contrast, our method explored some plot commonsense, compensating for hidden clues not mentioned in the text and voice. It can explicitly model subtle conflict by a chain, thereby being able to make correct judgments. More cases were given in Appendix C.4.

4 Related Work

Sarcasm detection has a wide range of applications in consumer opinion analysis (Wang et al., 2026), empathetic dialogue (Chang et al., 2026), emotion education, etc. Early research mainly focused on sarcasm in text based on features like sentiment words (Ghosh & Veale, 2017), punctuation (Joshi et al., 2017), and *POS* tags (Maynard & Greenwood, 2014). Some works investigated the emotion in voice via prosodic features such as pitch, speech rate, and stress (Riloff et al., 2013). These features were hand-crafted, having poor scalability. Researchers gradually turned to neural networks, which were more flexible and can learn predictive patterns from data. To better capture the context, some work incorporated social features such as reply and mention relations (Oprea & Magdy, 2020), or cross-modal features (Cai et al., 2019). They applied alignment (Li et al., 2024) or graph-based networks (Liang et al., 2022) to grasp relations between features in different modalities. However, they were hard to capture implicit sarcasm, i.e., those with no explicit ironic words or tones. Another study proposed to resort to external knowledge (Castro et al., 2019), but they did not make good use of it. They either treated *LLMs* as a large knowledge source (Yao et al., 2025) and designed prompts (Ray et al., 2022) as guidance to retrieve knowledge for satire detection (Wang et al., 2024), or simply injected all knowledge into the model (Wei et al., 2025), or built an entity graph (Liang et al., 2024) with enhanced knowledge (Qiao et al., 2023). For this approach, it is hard to explain why it is satire. Besides, it is prone to being disturbed by redundant and contradictory noises in sarcastic expressions, leading to misjudgments. Moreover, implicit sarcasm rarely relies on a single contradiction. Instead, it often involves multiple layers of constraints. The above methods are hard to detect these indirect emotional conflicts. In addition, some work addressed other issues like class imbalance (Qin et al., 2023), feature confusion (Qiu et al., 2025), missing modalities (Jia et al., 2024), low-resource detection (Alnajjar & Hämäläinen, 2021), and sarcasm intensity (Van Hee et al., 2018). Differently, we focus on detecting implicit sarcasm in speech, with complex hidden conflicts.

5 Conclusion

This paper proposed a valuable task for detecting implicit sarcasm in speech. The sarcastic expression was hidden and inconsistent with the true intention. It was hard to detect this conflict via shallow matching of features in a single modality. Thus, we proposed a new method by using complementary context from multiple sources, i.e., the voice and transcript text, and external knowledge. That helped build an emotion graph to fully capture potential subtle correlations. By high-order reasoning on the graph, we can accurately infer a conflict chain to sarcasm. Extensive experiments were conducted to evaluate our approach. The results showed the effectiveness of our method. Further, we created a large-scale dataset, i.e., *AEbSD*, that can promote the development of this research field.

AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing, grammatical correction, and limited code refinement. We did not use generative AI tools for research idea generation and methodology design. Any AI-assisted code modifications were restricted to implementation-level improvements, such as code restructuring and debugging support, without changing the underlying methodology. All AI-assisted revisions were manually reviewed and verified by the authors to ensure technical accuracy and consistency with the original content.

ETHICS STATEMENT

This paper proposed a new method that can well detect implicit sarcasm. It serves as an auxiliary tool that can be used to assist in voice content understanding and quality control on social platforms or communities. This technology can identify potentially misleading or conflicting satirical content, helping young listeners accurately grasp the true attitude. In addition, our model has an explainable design, allowing the tracing of the satire clues. This can facilitate users in verifying the rationality of results, eliminating the unfair risks brought by *black box decision-making*. The training and optimization process of our model strictly avoids discriminatory characteristics such as gender, age, region, and religion. The collection and usage of these datasets followed the open-source and copyright agreements of their respective publishers. The detection is based solely on the satirical features of speech, not targeting specific groups or individuals. The design of this technology and its applications strictly follow ethical norms, and there are no substantive moral risks.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Khalid Alnajjar and Mika Hämäläinen. ! qu\'e maravilla! multimodal sarcasm detection in spanish: a dataset and a baseline. *arXiv preprint arXiv:2105.05542*, 2021.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, pp. 1–12, 2020.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. Whisperx: Time-accurate speech transcription of long-form audio. *arXiv preprint arXiv:2303.00747*, 2023.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable pertubed optimizers. *in Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, pp. 9508–9519, 2020.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, pp. 335–359, 2008.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2506–2515, 2019.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. Towards multimodal sarcasm detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4619–4629, 2019.

Jeremy Chang, Chia-Yun Hsu, and Chung-Hsien Wu. Detecting sarcasm in multiple modalities by leveraging sarcasm type correlations. *IEEE Access*, 14:3606–3617, 2026.

Kai-Wei Chang, Yu-Kai Wang, Hua Shen, Iu-thing Kang, Wei-Cheng Tseng, Shang-Wen Li, and Hung-yi Lee. Speechprompt v2: Prompt tuning for speech classification tasks. *arXiv preprint arXiv:2303.00733*, 2023.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*, 2023.

Zeyu Deng, Yanhui Lu, Jiashu Liao, Shuang Wu, and Chongfeng Wei. Sync-tva: A graph-attention framework for multimodal emotion recognition with cross-modal fusion. *arXiv preprint arXiv:2507.21395*, 2025.

Junlin Fang, Wenya Wang, Guosheng Lin, and Fengmao Lv. Sentiment-oriented sarcasm integration for video sentiment analysis enhancement with sarcasm assistance. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 5810–5819, 2024.

Aniruddha Ghosh and Tony Veale. Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 482–491, 2017.

Yuan Gong, Alexander H Liu, Hongyin Luo, Leonid Karlinsky, and James Glass. Joint audio and speech understanding. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 1–8, 2023.

Mengzhao Jia, Can Xie, and Liqiang Jing. Debiasing multimodal sarcasm detection with contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18354–18362, 2024.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. Automatic sarcasm detection: A survey. *Journal of ACM Computing Surveys*, pp. 1–22, 2017.

Fan Li, Jianxing Yu, Jielong Tang, Wenqing Chen, Hanjiang Lai, Yanghui Rao, and Jian Yin. Answering complex geographic questions by adaptive reasoning with visual context and external commonsense knowledge. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 25498–25514, 2025.

Mingcheng Li, Dingkang Yang, Xiao Zhao, Shuaibing Wang, Yan Wang, Kun Yang, Mingyang Sun, Dongliang Kou, Ziyun Qian, and Lihua Zhang. Correlation-decoupled knowledge distillation for multimodal sentiment analysis with incomplete modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12458–12468, 2024.

Bin Liang, Chenwei Lou, Xiang Li, Min Yang, Lin Gui, Yulan He, Wenjie Pei, and Ruifeng Xu. Multi-modal sarcasm detection via cross-modal graph convolutional network. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 1767–1777, 2022.

Bin Liang, Lin Gui, Yulan He, Erik Cambria, and Ruifeng Xu. Fusion and discrimination: A multimodal graph contrastive learning framework for multimodal sarcasm detection. *IEEE Transactions on Affective Computing*, pp. 1874–1888, 2024.

Diana G Maynard and Mark A Greenwood. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pp. 4238–4243, 2014.

Gaspard Michel, Elena V Epure, and Christophe Cerisara. Computational narrative understanding for expressive text-to-speech. In *Findings of the Association for Computational Linguistics*, pp. 6194–6215, 2026.

Silviu Oprea and Walid Magdy. isarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1279–1289, 2020.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 31–41, 2016.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 527–536, 2019.

Yang Qiao, Liqiang Jing, Xuemeng Song, Xiaolin Chen, Lei Zhu, and Liqiang Nie. Mutual-enhanced incongruity learning network for multi-modal sarcasm detection. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 9507–9515, 2023.

Libo Qin, Shijue Huang, Qiguang Chen, Chenran Cai, Yudi Zhang, Bin Liang, Wanxiang Che, and Ruifeng Xu. Mmsd2. 0: Towards a reliable multi-modal sarcasm detection system. *arXiv preprint arXiv:2307.07135*, 2023.

Ziqi Qiu, Jianxing Yu, Yufeng Zhang, Hanjiang Lai, Yanghui Rao, Qinliang Su, and Jian Yin. Detecting emotional incongruity of sarcasm by commonsense reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 9062–9073, 2025.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 28492–28518, 2023.

Anupama Ray, Shubham Mishra, Apoorva Nunna, and Pushpak Bhattacharyya. A multimodal corpus for emotion recognition in sarcasm. *arXiv preprint arXiv:2206.02119*, 2022.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 704–714, 2013.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: A family of highly capable multimodal models. arxiv 2023. *arXiv preprint arXiv:2312.11805*, 2024.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pp. 39–50, 2018.

Anthony J Viera, Joanne M Garrett, et al. Understanding interobserver agreement: the kappa statistic. *In Journal of Fam med*, 37(5):360–363, 2005.

Peng Wang, Yongheng Zhang, Hao Fei, Qiguang Chen, Yukai Wang, Jiasheng Si, Wenpeng Lu, Min Li, and Libo Qin. S3 agent: unlocking the power of vllm for zero-shot multi-modal sarcasm detection. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21 (11):1–16, 2024.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *Proceedings of the World Wide Web Conference*, pp. 2022–2032, 2019.

Yihua Wang, Qi Jia, Cong Xu, Feiyu Chen, Yuhan Liu, Haotian Zhang, Liang Jin, Lu Liu, and Zhichun Wang. Conditional information bottleneck for multimodal fusion: Overcoming shortcut learning in sarcasm detection. *arXiv preprint arXiv:2508.10644*, 2025.

Yihua Wang, Qi Jia, Cong Xu, Feiyu Chen, Yuhan Liu, Haotian Zhang, Liang Jin, Lu Liu, and Zhichun Wang. Conditional information bottleneck for multimodal fusion: Overcoming shortcut learning in sarcasm detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26652–26660, 2026.

Yiwei Wei, Hengyang Zhou, Shaozu Yuan, Meng Chen, Haitao Shi, Zhiyang Jia, Longbiao Wang, and Xiaodong He. Deepmsd: Advancing multimodal sarcasm detection through knowledge-augmented graph reasoning. *IEEE Transactions on Circuits and Systems for Video Technology*, 35 (7):6413–6423, 2025.

Ben Yao, Yazhou Zhang, Qiuchi Li, and Jing Qin. Is sarcasm detection a step-by-step reasoning process in large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 25651–25659, 2025.

Meishan Zhang, Yue Zhang, and Guohong Fu. Tweet sarcasm detection using deep neural network. In *Proceedings the 26th International Conference on Computational Linguistics*, pp. 2449–2460, 2016.

Jiaqi Zhao, Fei Wang, Kun Li, Yanyan Wei, Shengeng Tang, Shu Zhao, and Xiao Sun. Temporal-frequency state space duality: An efficient paradigm for speech emotion recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2025.

Miao Zhou, Lina Yang, Thomas Wu, Dongnan Yang, and Xinru Zhang. Dual-path dynamic fusion with learnable query for multimodal sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 11366–11376, 2025.

Yan Zhuang, Yanru Zhang, Zheng Hu, Xiaoyue Zhang, Jiawen Deng, and Fuji Ren. Glomo: Global-local modal fusion for multimodal sentiment analysis. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 1800–1809, 2024.

A Extra Details of Methodology

A.1 Prompt of Knowledge Augmentation

We design prompts that can guide the *LLMs* to better understand our tasks and to generate emotional knowledge clues that are helpful for reasoning.

Prompt 1: Structured Knowledge Generation for Implicit Sarcasm Detection in Speech

[SYSTEM ROLE & OBJECTIVE]

You are a professional knowledge generation engine dedicated to implicit sarcasm detection research. Your core objective is to generate two types of structured knowledge — **textual commonsense knowledge** and **prosodic emotional knowledge** — based on the input speech, providing interpretable reasoning support for implicit sarcasm detection. All outputs must be directly traceable to the provided input features and must comply with academic rigor.

[TASK DEFINITION]

You will accept three types of inputs and produce two structured knowledge modules:

- **Inputs**: (i) textual features; (ii) prosodic features (speaker-specific attributes and interfering noise removed, retaining only key prosodic clues that reflect emotional expression); (iii) retrieval-augmented external context (sorted by relevance to the text, higher-relevance entries appear first).
- **Output Module A — Text Commonsense Knowledge**: factual commonsense knowledge strongly aligned with the textual context (e.g., scenario conventions, lexical/metaphorical meanings) that resolves semantic ambiguity and supports sarcasm reasoning at the semantic level.
- **Output Module B — Prosodic Emotional Knowledge**: mapping relations between prosodic features and emotional tendencies that complement acoustic clues for sarcasm reasoning at the acoustic-emotion level.

[INPUT / OUTPUT SPECIFICATION]

Input format (feature embeddings, delimited as follows):

<s> textual features <s> retrieval-augmented content <s> prosodic features

Output format: a single JSON object with exactly two top-level fields, structured as below.

```
{
  "text_commonsense": {
    "content": "<factual commonsense supporting sarcasm reasoning,
                e.g., scenario conventions or lexical metaphors;
                <= 200 words>",
    "confidence": <float in [0, 1], scored by factuality
                   and semantic relevance to the text>
  },
```

```
  "prosodic_emotional": {
    "content": "<follow the three-element chain:
                Specific Prosodic Indicator
                $\to$ Corresponding Emotion
                $\to$ Matching Logic;
                <= 200 words>",
    "confidence": <float in [0, 1], scored by academic
                   consensus on acoustic-emotion matching>
  }
}
```

[CONSTRAINTS]

- **Factuality**: all knowledge content must be objectively factual; subjective speculation is strictly prohibited.
- **Grounding**: text commonsense must be closely aligned with the textual and retrieved external context; prosodic emotional knowledge must comply with established academic norms for acoustic emotion recognition.
- **Evidence basis**: no knowledge shall be generated without a traceable basis in the provided input evidence.
- **Confidence calibration**: confidence scores must reflect academic consensus, not arbitrary assignment.
- **Conciseness**: redundant content is forbidden; each knowledge field is strictly limited to **200 words or fewer**.

A.2 Details of Graph Refinement

We refine the initial emotion graph to reduce its potential redundancy and noise. In detail, we first take each node v as the center and calculate its edge weight $\omega_{v,i}^{\Phi}$ based on meta-paths from *text token* to *prosodic segment* and *prosodic knowledge*. This weight is used to reduce spurious correlations. We then explore a linear programming strategy based on this $\omega_{v,i}^{\Phi}$ to select useful edges. The strategy has good global optimization ability. It can help to avoid mis-deletion caused by outliers, ensuring the stability and rationality of the results. Two constraints are adopted, including (1) each node retains only l associated edges, preventing scattered associations of nodes; (2) the weights of the retained l edges should be higher than those of the unselected edges, ensuring the monotonicity of selection. To derive the top-l edges, we need to make discrete "*reserve/delete*" decisions, which are non-differentiable and hard to optimize. To tackle it, we employ the maximum perturbation technique, that is, adding *Gaussian* noise $\mathbb{U} \sim N(0, \sigma^2)$. This can transform the discrete operation into a continuous distribution $\mathcal{S}_p^{ij}$. $\mathcal{S}_p^{ij}$ represents the probability of a connection between node i and j. The continuous $\mathcal{S}_p^{\vartheta}$ can be updated via gradient backpropagation, enabling end-to-end training. This allows the graph topology to adapt to subsequent reasoning modules. Based on $\mathcal{S}_p^{ij}$, we form an adjacency matrix $\mathcal{A}_{ij} = \mathcal{S}_p^{ij} \times \omega_{ij}^{\Phi}$. We retain connections with high probability $\mathcal{S}^{ij}$ and high edge weight (ω_{ij}^{Φ}). Respectively, we weaken the ones with low probability or weight. This can derive a refined graph with a good topological structure to well support high-order reasoning.

A.3 Details of Conflict Inference

State Space: At each reasoning step z, the state s_z jointly encoded the local context of current node and the global topology of the subgraph $\mathcal{G}_s^i$, as $s_z = f_{\text{enc}}\big(h_z^{node}, \mathcal{A}_s\big)$, where $h_z^{node} \in \mathbb{R}^d$ was the embedding of the current node v_z, h_z^{node} was obtained from the heterogeneous graph encoder. $\mathcal{A}_s \in \mathbb{R}^{|\mathcal{V}_s| \times |\mathcal{V}_s|}$ was the weighted adjacency matrix of $\mathcal{G}_s^i$. It was used to encode the edge and its type. The encoder f_{enc} was a single-layer graph attention. It diffused global structural context into h_z^{node}, enabling the deduction of the neighborhood of v_z based on the graph topology.

Action Space: At step z, the action space was the candidate adjacency set of the current node, $\mathcal{A}_z = \mathcal{N}(v_z) = \{v \in \mathcal{V}_s \mid (v_z, v) \in \mathcal{E}_s\}$, where $\mathcal{E}_s$ denoted the edge set of $\mathcal{G}_s^i$. The state $s_z \in \mathbb{R}^{256}$ was the output of a graph attention network. It encoded the current node v_z and its local context. To incorporate knowledge reasoning capabilities, s_z was further conditioned on the hidden representation extracted from *LLaMA3-8B* (with *LoRA* adapters). It was projected into the same 256-d space via a learnable linear layer before being fed into the policy network. The

network produced a probability distribution over candidates as $\pi_\theta(a_z \mid s_z) = \text{softmax}(W_a\, s_z)$, where $W_a \in \mathbb{R}^{|\mathcal{A}_z| \times 256}$ was a learnable type-aware matrix. During trajectory rollout, actions were sampled stochastically from $\pi_\theta(\cdot \mid s_z)$ for exploration. At inference time, the action was selected greedily as $a_z^* = \arg\max_{a \in \mathcal{N}(v_z)} \pi_\theta(a \mid s_z)$. The reasoning chain was initialized from the word node of the most query-relevant token. It was selected based on the attention score of all word nodes in $\mathcal{G}_s^i$.

Termination Condition: The trajectory τ_z terminated when any of three conditions was satisfied, including (1) *Maximum horizon*: the step count had reached the predefined upper bound z_{max}, preventing excessively yielding too-long or redundant chains; (2) *Prosody node arrival*: every valid reasoning chain should have full conflicts between the word, prosody and external knowledge. The sarcastic incongruity can be inferred with a certain acoustic evidence node; (3) *Low-reward early stopping*: when the reward r_z fell below a threshold ρ, the current trajectory is highly likely to have deviated from sarcasm-related regions in the graph. A good terminal step was $z_T = \min(z_{max},\, z_{\text{prosody}},\, z_\rho)$. Its discounted return was computed as $R_z = \sum_{l=0}^{z_T - z} \gamma^l\, r_{z+l}$.

Reward Function: To yield optimal reasoning paths, we defined the reward r_z by a weighted combination of three metrics, including (1) *Node-query semantic similarity* ($r_{\text{node},z}$). This term quantified the semantic relevance between the current node and a task-specific query vector h_{query}. It was constructed in four steps. Given the input voice a, we first derived its content text t and prosodic features. Their encodings H_t and H_a were fused as $H_{mul} = \hat{\alpha}\, H_t + (1 - \hat{\alpha})\, H_a$, where $\hat{\alpha} \in [0, 1]$. Given the textual commonsense set $\{H_{ct_1}, \ldots, H_{ct_m}\}$ and the prosodic emotion knowledge set $\{H_{kp_1}, \ldots, H_{kp_n}\}$, we then computed their respective means, $H_{ct} = \frac{1}{m}\sum_{i=1}^{m} H_{ct_i}$, $H_{kp} = \frac{1}{n}\sum_{j=1}^{n} H_{kp_j}$, and fused them as $H_{\text{ag}} = \hat{\beta}\, H_{ct} + (1 - \hat{\beta})\, H_{kp}$, where $\hat{\beta} \in [0, 1]$. Next, we projected the fused representation into a learnable matrix $\mathcal{M}_{task}$ and conducted element-wise interaction with H_{mul}: $H_{\text{tmp}} = \mathcal{M}_{task}\,(H_{mul} \odot H_{ag})$. Finally, we applied ℓ_2 normalization to standardize the scale, as $h_{\text{query}} = \ell_2(H_{\text{tmp}})$. Based on h_{query}, the node-query reward at step z was $r_{\text{node},z} = \sigma(\cos(h_{s_z},\, h_{\text{query}})) \cdot w_{\text{type}}(s_z)$, where h_{s_z} is the ***BERT*** embedding of node s_z, $\sigma(\cdot)$ is the sigmoid function, and $w_{\text{type}}(s_z) \in [1.2, 1.5]$ was a type-specific gain for conflict, it was set to 1 for irrelevant nodes. h_{query} was constructed only once from the episode-level inputs prior to trajectory rollout. It was then fixed per episode, but $r_{\text{node},z}$ can still vary across steps since the current node s_z changes during traversal. This design prioritized those nodes that can reflect sarcastic conflicts or provide background knowledge during exploration. (2) *Edge-level semantic consistency* ($r_{\text{edge},z}$). It evaluated the rationality of the state transition from s_{z-1} to s_z based on the edge weight $w(a_z)$. It reflected the semantic affinity between adjacent nodes. A higher weight indicated stronger inter-node association and more coherent reasoning logic, as $r_{\text{edge},z} = \frac{w(a_z) - w_{\min}}{w_{\max} - w_{\min}}$, where $w_{\min}$ and $w_{\max}$ were the minimum and maximum weights in $\mathcal{G}_s^i$, respectively, ensuring comparability across subgraphs of varying scale. (3) *Cross-modal alignment* ($r_{\text{cross},z}$). This term encouraged transitions across heterogeneous modalities, i.e., from a textual or commonsense node to a prosody node, since that was the sarcastic incongruity place. Since prosody nodes encoded acoustic features rather than visual content, we projected the source node $h_{s_{z-1}}$ and the target prosody h_{s_z} into a shared latent space to measure their similarity, as Eq.(13), where ϕ_{src} and ϕ_{tgt} were modality-specific linear projections that can be trained end-to-end with the policy network.

$$
\begin{aligned}
u_{z-1} = \phi_{\text{src}}\big(h_{s_{z-1}}\big)\,, v_z = \phi_{\text{tgt}}(h_{s_z})\,, \\
r_{\text{cross},z} = \begin{cases} \sigma(\cos(u_{z-1}, v_z))\,, & \text{if cross-modal,} \\ 0, & \text{otherwise.} \end{cases}
\end{aligned} \tag{13}
$$

Warm-Up Subgraph Sampling: To initialize the policy network with basic graph traversal patterns before full RL training, we constructed a warm-up dataset via a subgraph sampling procedure. For each training instance, a connected subgraph $\mathcal{G}_s^i$ of fixed size K was sampled from the full graph $\hat{\mathcal{G}}$ by two stages. We first identified all cross-modal edges of type *knowledge–prosody* in $\hat{\mathcal{G}}$, as these edges were the structural loci of sarcastic incongruity. One such edge (v_c, v_p) was randomly selected as the anchor, where v_c was a knowledge node and v_p was a prosody node. That ensured every sampled subgraph contained at least one cross-modal bridge, providing a meaningful cross-modal conflict signal for training. Afterward, starting from the anchor edge (v_c, v_p), we performed a breadth-first expansion over $\hat{\mathcal{G}}$ until the subgraph reached exactly K nodes. We then retained all edges among the collected nodes. The resulting $\mathcal{G}_s^i$ was ensured to be connected and cover the full reasoning progress

over words, knowledge, and prosody, etc. During each warm-up iteration, a subgraph $\mathcal{G}_s^i$ was sampled randomly via this procedure, preventing overfitting.

B Details of *AEbSD* Dataset

We illustrated the dataset statistics, construction pipeline, annotation protocols, inter-annotator agreement, and ethical considerations, as follows.

B.1 Dataset Statistics and Distribution

The dataset statistics are shown in Table 3. To avoid bias, the label distribution was nearly balanced. For each sample, audio durations ranged from approximately 2 to 18 seconds, with a mean of 6.3 ± 3.1 seconds. Transcript text ranged from 5 to 60 tokens, with a mean of 18.4 ± 9.2 tokens.

Table 3: Statistics of *AEbSD* dataset across data splits.

Split	Total	Sarcastic	Non-Sarcastic	Ratio (Sarc.:Non.)
Train	7,584	3,792	3,792	1:1
Validation	2,167	1,084	1,083	≈1:1
Test	1,083	541	542	≈1:1
Total	**10,834**	**5,417**	**5,417**	**1:1**

B.2 Dataset Construction Pipeline

There were four stages to construct *AEbSD*.

Stage 1: Data Sourcing and Audio-Text Alignment. Audio recordings were sourced from the website *LibriVox*[1]. That is a volunteer-driven repository of public-domain audiobooks. The corresponding transcripts were retrieved from *Project Gutenberg*[2]. Both repositories were copyright-free and operated under public-domain and *Creative Commons Zero* (*CC0*) licenses, ensuring full compliance with legal and ethical data-use standards. We downloaded a total of 247 audiobooks, covering diverse genres, including fiction, classic literature, and drama. All of which were rich in ironic, sarcastic, and figurative expressions. Moreover, we employed *WhisperX* (Bain et al., 2023) to yield sentence-level voice-text pairs based on the timestamps. *WhisperX* combined *OpenAI Whisper* (Radford et al., 2023) for speech recognition with *wav2vec 2.0* (Baevski et al., 2020) for phoneme-level alignment, thereby producing a good segmentation pair. Each segment was represented as a tuple $\langle \mathbf{a}_i, t_i, \mathrm{ts}_i \rangle$, where $\mathbf{a}_i$ was the audio clip, t_i was the aligned transcript, and ts_i was the start/end timestamps. Segments shorter than 1 second or containing more than 30% transcription mismatch (measured by character-level edit distance against the *Gutenberg* ground truth) were discarded, yielding a candidate pool of approximately 30,000 pairs.

Prompt 2: LLM Candidate Scoring

```
You are an expert in spoken-language pragmatics and speech quality assessment. Your
task is to evaluate a voice-text pair extracted from a speech recording and assign
it a composite quality score for sarcasm dataset construction.

You will be given:
 • [Transcript]: The verbatim text of the speech segment.
 • [Context]: Up to five surrounding sentences from the source text.
 • [Audio Description]: A brief description of the speech properties (duration,
   estimated SNR, detected prosodic features).

Please evaluate the pair on the following three criteria and assign a sub-score from
0 (very poor) to 10 (excellent) for each:
```

[1] https://librivox.org/

[2] https://www.gutenberg.org/

(1) Prosodic Clarity [0–10]: Does the spoken delivery exhibit discernible prosodic cues (e.g., exaggerated intonation, stress, tempo modulation, tonal incongruity) that could indicate non-literal or sarcastic intent? A high score indicates clear, interpretable prosodic variation. A low score indicates flat, monotonous, or indistinguishable delivery.

(2) Signal-to-Noise Ratio [0–10]: Is the audio recording sufficiently clean for reliable acoustic feature extraction? A score of 10 indicates studio-quality audio with no background noise. Penalise for audible background noise, echo, clipping, or overlapping speech.

(3) Contextual Coherence [0–10]: Does the text segment form a semantically interpretable unit within its narrative context? Can the sarcastic or non-sarcastic intent be reasonably assessed from this segment (with the provided context)? A high score indicates a self-contained, contextually grounded utterance. A low score indicates an elliptical fragment, an incomplete sentence, or one that is entirely unintelligible without extensive additional context.

Output your evaluation in the following JSON format:

```
{
  "prosodic_clarity": <int, 0-10>,
  "snr": <int, 0-10>,
  "contextual_coherence": <int, 0-10>,
  "composite_score": <float, average of the three>,
  "brief_rationale": "<one sentence explaining the composite score>"
}
```

Do NOT include any text outside the JSON object.

Stage 2: LLM-Based Candidate Pre-scoring. Considering the high cost of manual annotation, we customized a prompt (see 2) to guide the *LLM* (i.e., *GPT-4o* (Achiam et al., 2023)) to filter out obviously bad cases. For each candidate, *LLM* assigned a quality score s_i ranging from 0 to 10 based on three criteria. (1) *Prosodic clarity*: whether the speech voice possesses sufficient distinct pitch features, such as pitch variations, stress patterns, or rhythm changes. For each candidate, if its pitch features are unclear or obscured by recording defects, its score is lower. (2) *Signal-to-noise ratio (SNR)*: whether the voice is clean enough to extract acoustic features reliably. If it contains noise, overlaps, or poor recording quality, this score will be small. (3) *Contextual coherence*: whether the text contains ironic elements. A good sample contains three to five sentences to cover sufficient context for detection. If the text is too short or overly abbreviated, its score will be small. Candidates with a sum score s_i less than a threshold (i.e., 5) were removed. In this way, we collected about 15084 pairs in descending order of the score s_i.

Stage 3: Human Annotation and Quality Control. We recruited fifteen annotators who majored in linguistics and computer science. They were good at English. They were trained on this task, involving the definition of implicit sarcasm, positive and negative cases, and annotation criteria. Each independently labelled candidates as either *sarcastic* or *non-sarcastic* according to the two criteria, including (1) *Cross-modal incongruity.* The prosody or tone of the voice contradicts the surface words of the transcript. The cases detectable only from text surface features (e.g., ironic words) or acoustic clues (e.g., mocking tone) were downgraded for later processing. (2) *Knowledge dependency.* A good implicit sarcastic case should rely on background knowledge. Those can be easily identified without the need for such knowledge were viewed as non-implicit sarcastic and excluded.

If a candidate satisfied two criteria, it was viewed as sarcastic; otherwise, non-sarcastic. In addition, each case was cross-validated by three independent volunteers. Final labels were determined by majority voting.

Stage 4: Metadata Enrichment. Each sample was further enriched with two types of metadata derived from the source narrative, including (1) *Speaker identity*, i.e., the name and role of the speaking character, e.g. protagonist, antagonist, narrator, minor character, etc. They were acquired from the *Gutenberg* source. (2) *Inter-character relations*, the social, emotional, or power-dynamic relations between the character and the one with whom the utterance occurs (e.g., *"rivals," "romantic partners," "employer-employee"*). These metadata were extracted via *NER* tool (i.e., `spaCy`) and manually verified. They can provide a better context for detection (Michel et al., 2026).

Annotation Consistency. To quantify the label quality, we computed *Cohen's Kappa* (Oraby et al., 2016) κ for 500 instances (250 positive, 250 negative) which were randomly sampled. The κ was 0.81, indicating a good agreement (Oprea & Magdy, 2020). Disagreements were most common in cases where cross-modal incongruity was mild or the requisite background knowledge was distributed unevenly among annotators.

Ethical Considerations. Each annotator received \$0.05 per sample, yielding an approximate hourly wage of \$2.50 at a rate of 50 samples per hour. Their compensation meets or exceeds their local minimum wage requirement. Annotators were informed of the voluntary nature of their participation and were free to withdraw at any time without penalty. No personally identifiable information was collected from annotators.

For content sensitivity, the dataset contained sarcastic utterances drawn from classical and contemporary fiction. Some texts may contain language or social dynamics that reflected historical attitudes (e.g., class prejudice, gender stereotypes). These instances were retained as linguistically authentic examples of sarcasm. The dataset was designed exclusively for research purposes. It should not be used for any application that targets, discriminates against, or profiles individual speakers.

C Additional Evaluations

C.1 Implementation Settings

Our Method: The hyperparameter settings were summarized in Table 4. Our policy network π_θ was initialized from *LLaMA3-8B* with *LoRA* adapters (rank $r = 16$, scaling $\alpha_{\text{lora}} = 32$) applied to the query and value projection matrices of each attention layer. All other parameters were frozen during RL training. The value function head V_θ was a two-layer *MLP* (hidden size 512, ReLU) appended to the final hidden state. The graph encoder f_{enc} used a single-layer graph attention network with 8 heads and a hidden dimension $d = 256$. The modality projection layers ϕ_{src} and ϕ_{tgt} were single linear layers. They were mapped into a shared space of $d = 256$, trained end-to-end with the policy network. In the RL training, the warm-up phase ran for 3,000 steps with a linear learning rate schedule. *PPO* training ran up to 6,000 steps with a rollout buffer of 2,048 transitions per update and 4 mini-batch epochs per *PPO* iteration, with validation every 200 steps. Early stopping was triggered if no improvement in validation loss was observed for one consecutive round. All experiments were conducted on 4 $\times$ NVIDIA RTX 3090 GPUs (24 GB VRAM each) using mixed-precision training (`fp16`). Training the full model took approximately 7 hours; inference on 1,000 test samples completes in approximately 5 minutes.

SpeechPrompt v2: We adopted pre-trained *LMs* (i.e., *GSLM* and *pGSLM*) for speech prompt tuning. In addition, we employed the pre-trained *HuBERT* with 100 clusters. All pre-trained models, quantizers, and spoken *LLMs* are publicly accessible on *fairseq2*. We fixed the prompt length at $l = 5$. The verbalizer was implemented as a linear model that can map discrete features to sarcasm labels.

LTU-AS: We employed a three-step training. We first trained the randomly initialized projection layer on 2.1M samples. The learning rate was set to $1e^{-3}$ and trained for 2 epochs, with *TLTR* and *LoRA* adapters kept frozen. Next, we unfroze the *TLTR* and *LoRA* adapters, and then trained them jointly with the projection layer on the same 2.1M samples. The learning rate was $2e^{-4}$, with 2 training epochs. Finally, we trained the full parameter set on all tasks using 9.6M samples. The learning rate remained $2e^{-4}$, and we trained for 1 epoch.

MUStARD++: We utilized a *BART*-large model with a 1024-dimensional output to extract textual features. We processed 22.5KHz audio by splitting it into equal-length segments. We then extracted the *MFCC*, *Mel spectrogram* features via *Librosa* and prosodic features via *OpenSMILE*. We averaged across segments to form a voice vector. We conducted one-vs-rest experiments for each emotion. We used a *sigmoid* head layer instead of a *softmax* for the detection task. Hyperparameter search was performed for all training processes, including dropout rates $[0.2, 0.3, 0.4]$, learning rates $[0.001, 0.0001]$, batch sizes $[32, 64, 128]$, shared embedding sizes $[2048, 1024]$, and projection embedding sizes $[1024, 256]$.

PS2RI: The learning rates for the acoustic encoder were set to $3e^{-3}$. The language encoder and other blocks were set to $1e^{-5}$. We used the *Adam* optimizer and cosine decay policy. The batch size was set to 64, the dropout rate to 0.4, and the training epochs to 50, with an early stopping mechanism for

Table 4: Hyperparameter settings. Values marked with * are empirically selected.

Hyperparameter	**Symbol**	**Value**
Retrieval & Graph		
Retrieved contents	N	3
Relevance weight	δ	0.4
Knowledge threshold	$\tilde{\epsilon}$	0.95
Graph control parameter	ϑ	0.2
Local attention window	w	2^*
Rationality scaling coefficient	ξ	1.0^*
Graph similarity threshold	τ_g	0.7^*
Retained edges per node	l	5^*
Reinforcement Learning		
Discount factor	γ	0.99
GAE lambda	λ_{GAE}	0.95
PPO clip coefficient	ϵ_{clip}	0.2
Entropy coefficient	β_H	0.01
Value loss coefficient	c_b	0.3
Max trajectory length	$z_{\max}$	6
Early stopping threshold	ρ	0.1
Reward weights	$\lambda_1, \lambda_2, \lambda_3$	0.3, 0.3, 0.4
Node type gain	w_{type}	$[1.2, 1.5]$
Warm-up subgraph size	K	32
LoRA rank	r	16
LoRA scaling	α_{LoRA}	32
Graph hidden dimension	d	256
Optimization		
Optimizer	—	AdamW
Adam coefficients	β_1, β_2	0.9, 0.999
Adam epsilon	ϵ_{Adam}	10^{-8}
Weight decay	λ_{wd}	10^{-4}
LR (pre-trained modules)	η_{pre}	2×10^{-5}
LR (new modules)	η_{new}	1×10^{-4}
Gradient clip norm	$g_{\max}$	1.0
Warm-up steps	S_{warm}	3,000
Max training steps	$S_{\max}$	6,000
Orthogonality loss weight	λ_{dis}	0.1^*
Inconsistency sensitivity	κ_{inc}	1.0^*
Conflict-aware weighting coefficient	η_{con}	1.0^*
Sarcastic-sample balance coefficient	β_{inc}	1.0^*
Inconsistency margin	m	0.5^*
Overall loss weights	$\lambda_{\text{warm}}, \lambda_{\text{ppo}}, \lambda_{\text{sd}}$	0.2, 0.4, 0.4^*

overfitting mitigation. We employed a 12-layer *ALBERT*, 8-layer transformer, and 1-layer transformer to process text and acoustic modalities, respectively. We configured 4 *MTF* layers in the sarcasm feature encoder and 4 iterations for the model, with a hidden-layer dimension of 192.

Sync-TVA: We used *AdamW* as an optimizer, with a learning rate of 1×10^{-4}, batch size of 16. We employed a uniform *L2* regularization coefficient of 1×10^{-3} to reduce overfitting. The dropout rate was set to 0.2, the model layer count to 7, the position encoding dimension to 17, the attention head number to 19, and the loss function weights to 1.0 and 0.7, respectively.

GLoMo: We employed the *AdamW* optimizer with a fixed random seed of 5576. We set consistent hyperparameters across three experts in all *MoE* layers. The learning rate was tuned from $[1e^{-5}, 2e^{-5}, 4e^{-5}]$, and hidden layer dimensions were adjusted from $[48, 112, 160, 192, 256]$. In addition, the batch size was tuned from $[4, 8, 16]$, and the dropout rate was set to 0.2.

DPDF-LQ: The query length was set to 8, and its dimension was set to 128. The *DGLQA* depth was set to 3. The depth of *GPath CA* and *LPath CA* was tuned from $[2, 4]$. The hidden dimension was 256. The learning rate and the weight decay were both $1e^{-4}$, with a batch size of 64.

GPT-4o: We designed prompt as 3 based on the transcript text and the raw audio of each sample. We use it to guide the model to output a sarcasm label and a reasoning chain.

Prompt 3: Sarcasm Detection in Speech via GPT-4o

[SYSTEM ROLE & OBJECTIVE]
You are a professional sarcasm detection engine for speech utterances. Your objective is to determine whether a given utterance is sarcastic, based on both its transcript and audio, and to provide a structured reasoning chain that justifies your decision. All outputs must be grounded in the provided inputs and comply with academic rigor.

[TASK DEFINITION]
You will accept two types of inputs and produce a structured output with two fields:

- **Inputs:** (i) the transcript text of the utterance; (ii) the raw audio of the utterance.
- **Output Field A – Label:** a binary sarcasm label: `sarcastic` or `non-sarcastic`.
- **Output Field B – Reasoning Chain:** a step-by-step reasoning chain explaining your decision, covering: (a) key textual cues, (b) relevant commonsense context, and (c) how the prosody aligns or conflicts with the literal meaning.

[INPUT / OUTPUT SPECIFICATION]
Input format:
`<s> transcript text <s> audio`

Output format: a single JSON object structured as below.

```
{
  "label": "<sarcastic | non-sarcastic>",
  "reasoning_chain": {
      "text_cues": "<key sarcasm-related words or phrases;
                   <= 100 words>",
      "commonsense": "<commonsense knowledge contextualising
                     the utterance; <= 150 words>",
      "prosody_conflict": "<analysis of prosody--text alignment
                          or conflict; <= 150 words>"
  }
}
```

C.2 Evaluations on Reasoning Chains

To assess the reliability of our derived conflict chains, we studied 200 randomly selected cases. We compared against chains generated by three state-of-the-art multimodal *LLMs* that natively accepted both audio and text as input, including *GPT-4o*, *Qwen-Audio* (Chu et al., 2023), and *Gemini 1.5 Pro* (Team et al., 2024). For fairness, all of them were guided by Prompt 4 in a zero-shot manner.

Prompt 4: Sarcastic Reasoning Chain Generation for Implicit Sarcasm Detection in Speech

[SYSTEM ROLE & OBJECTIVE]
You are a professional reasoning chain generation engine dedicated to implicit sarcasm detection in speech. Your core objective is to generate a structured reasoning chain that identifies cross-modal sarcastic incongruities — specifically, mismatches between **textual meaning** and **prosodic delivery** — based on multimodal inputs. All outputs must be directly traceable to the provided input features and must comply with academic rigor. **[TASK DEFINITION]**

You will accept three types of inputs and produce a structured three-step reasoning chain:

- **Inputs:** (i) the raw utterance text; (ii) a natural-language description of the speaker's prosodic characteristics (e.g., tone, pitch, speaking rate, and emotional expression), derived from the accompanying audio; (iii) retrieval-augmented external commonsense context (sorted by relevance to the text, higher-relevance entries appear first).
- **Step A – Word-Level Cues:** identify the key words or phrases in the text that carry potential sarcastic meaning, grounded in the textual content and commonsense context.
- **Step B – Commonsense Contextualisation:** provide relevant commonsense knowledge that contextualises the utterance and resolves semantic ambiguity at the *semantic level*.
- **Step C – Prosody–Text Conflict:** analyse how the speaker's prosody (tone, pitch, speaking rate) aligns or conflicts with the literal textual meaning, reasoning at the *acoustic-emotion level*.

[INPUT / OUTPUT SPECIFICATION]

Input format (plain text and audio, delimited as follows):
`<s> utterance text <s> prosodic description <s> retrieval-augmented content`
Output format: a single JSON object with exactly three top-level fields, structured as below.

```
{
  "step_a_word_cues": {
      "content": "<key sarcasm-related words or phrases
                  grounded in textual content;
                  <= 100 words>",
  },
  "step_b_commonsense": {
      "content": "<commonsense knowledge contextualising
                  the utterance at the semantic level;
                  <= 150 words>",
  },
  "step_c_prosody_conflict": {
      "content": "<analysis of prosody--text alignment
                  or conflict at the acoustic-emotion
                  level; <= 150 words>",
  }
}
```

This zero-shot prompt was used for three *LLM* baselines (*GPT-4o*, *Qwen-Audio*, and *Gemini 1.5 Pro*) to yield reasoning chains for human evaluation. Three annotators with good English independently rated each chain in terms of three metrics (1 = pass, 0 = fail), including (1) *Path completeness (PC):* whether the chain covered the clues from multiple modalities in *word*, *external knowledge*, and *prosody*. A chain that terminated prematurely before reaching a ground-truth step was marked as incomplete. (2) *Conflict validity (CV):* whether there is a cross-modal sarcastic incongruity. (3) *Semantic coherence (SC):* whether adjacent steps in the chain were logically correlated. The inter-annotator agreement was measured by κ. It was 0.79, indicating a substantial agreement (Viera et al., 2005). The disagreed cases were labeled by majority vote.

Table 5: Extra Human evaluation.

Method	**PC**	**CV**	**SC**
GPT-4o (zero-shot)	80.5	74.0	84.0
Qwen-Audio (zero-shot)	75.5	69.0	80.5
Gemini 1.5 Pro (zero-shot)	78.0	71.5	82.5
Ours	**89.5**	**85.0**	**95.0**

As shown in Table 5, our method outperformed three multimodal *LLM* baselines across all metrics. Among the baselines, *GPT-4o* achieved the highest scores, followed by *Gemini 1.5 Pro*, while *Qwen-Audio* lagged slightly behind. The largest margins over *GPT-4o* appeared on *SC* and *CV* (both +11.0%). That indicated that the graph structure can regularize the reasoning space and suppress incoherent transitions. The $r_{\text{cross},z}$ reward was designed for cross-modal mismatches. These mismatches were often overlooked for the general *LLMs*. The *PC* gap (+9.0%) further reflected that zero-shot baselines did not guarantee the traversal of the words, external knowledge, and prosody. But this traversal can be enforced by our method. Overall, 85% of our chains satisfied all three criteria, indicating good quality.

C.3 Evaluations of *LLM* Retrieved Knowledge

Moreover, we studied the quality of the retrieved knowledge from *LLMs*. We recruited 20 participants to analyze 700 randomly selected cases. Each scored each piece of retrieved knowledge in terms of five metrics, including *Grammatical correctness (GC)*, *Language fluency (LF)*, *Factual accuracy (FA)*, *Emotional relevance (ER)*, and *Input relevance (IR)*. The score was on a scale of 1 to 10, 10 indicating the highest quality. Each sample was rated by at least 4 participants. The kappa κ was 0.81, indicating a good agreement. We averaged all scores for each text-voice pair to determine its quality. As presented in Fig.(3c), *LlaMA-Omni* demonstrated a strong ability to understand textual content and prosodic features. It could supplement appropriate contextual clues to facilitate the inference of implicit sarcasm. In addition, we observed the suboptimal *LF* generated by *LlaMA* did not compromise the performance. That would be attributed to our graph reasoning module, which

anchored the inference logic by *word - knowledge - prosody* correlations, helping to mitigate the impact of *LF*.

C.4 Extra Case Studies

To better understand our model, we presented some bad cases. As shown in Table 6, our model erroneously retrieved the content "*cunning fox*". That is, foxes are usually cunning. That contradicted the character setting of a clumsy, good-natured fox in the given fictional story. This mismatch introduced a spurious node into the emotion graph, constructing a false conflict chain between the positively valenced text and the retrieved deceptive content. Although the prosody conveyed genuine warmth with no sarcastic clues, the model consequently misclassified a sincere compliment as sarcasm. It is worth noting that such niche fictional settings account for a small proportion in practical applications. The failure mode was not common in the dataset. It does not harm the overall performance and generalization ability of the model.

Table 6: A bad case (a situation that is rare). GT = ground-truth; Pred = predicted label.

Field	**Content**
Text	"What a remarkable leap! You are truly the most graceful creature in this forest."
Speaker	A clumsy, good-natured fox (story-specific setting)
Prosody	Flat, warm, and steady delivery; no pitch rise, no vocal tension, moderate speaking rate
GT Label	Non-sarcastic (sincere praise)
Pred Label	Sarcastic (incorrect)
Retrieved Commonsense	"Foxes are cunning and manipulative; their compliments often serve deceptive purposes." (*ConceptNet*)
Constructed Conflict Chain	*"remarkable leap" / "most graceful"* (positive text) → fox archetype: flattery as deception (commonsense) → flat prosody: no genuine emotional warmth (prosody) → inferred as insincere praise → predicted: sarcastic
Root Cause	The story defines the fox as clumsy and sincere, directly contradicting the universal *"cunning fox"* archetype. The retrieved commonsense prior overrides the story-specific setting, causing a spurious conflict chain that misclassifies a genuine compliment as sarcasm.