RAM-SD: Retrieval-Augmented Multi-agent framework for Sarcasm Detection

Ziyang Zhou*, **Ziqi Liu***, **Yan Wang, Yiming Lin and Yangbin Chen**†
Xi'an Jiaotong–Liverpool University
{ziyang.zhou22, ziqi.liu22, yan.wang2202, yiming.lin21, yangbin.chen}@xjtlu.edu.cn

Abstract

Sarcasm detection remains a significant challenge due to its reliance on nuanced contextual understanding, world knowledge, and multi-faceted linguistic cues that vary substantially across different sarcastic expressions. Existing approaches, from fine-tuned transformers to large language models, apply a uniform reasoning strategy to all inputs, struggling to address the diverse analytical demands of sarcasm. These demands range from modeling contextual expectation violations to requiring external knowledge grounding or recognizing specific rhetorical patterns. To address this limitation, we introduce **RAM-SD**, a **R**etrieval-**A**ugmented **M**ulti-Agent framework for **S**arcasm **D**etection. The framework operates through four stages: (1) contextual retrieval grounds the query in both sarcastic and non-sarcastic exemplars; (2) a meta-planner classifies the sarcasm type and selects an optimal reasoning plan from a predefined set; (3) an ensemble of specialized agents performs complementary, multi-view analysis ; and (4) an integrator synthesizes these analyses into a final, interpretable judgment with a natural language explanation. Evaluated on four standard benchmarks, RAM-SD achieves a state-of-the-art Macro-F1 of 77.74%, outperforming the strong GPT-4o+CoC baseline by **7.01** points. Our framework not only sets a new performance benchmark but also provides transparent and interpretable reasoning traces, illuminating the cognitive processes behind sarcasm comprehension.

1 Introduction

Sarcasm detection remains a formidable challenge in natural language processing due to the intricate interplay of context, world knowledge, and multi-faceted linguistic cues required for its comprehension (Joshi et al., 2017). The ability to accurately identify sarcastic intent is crucial for developing more robust human-computer interaction systems and for enabling deeper analysis of social media content, from public opinion mining to user modeling. However, the sheer diversity of sarcastic expression, which ranges from simple verbal irony to complex, culturally-grounded statements, presents a significant hurdle for automated systems (Riloff et al., 2013). Early research sought to capture sarcastic signals through feature engineering (Davidov et al., 2010) and conventional deep learning architectures (Ghosh et al., 2017). More recently, the advent of Pre-trained Language Models such as BERT (Devlin et al., 2019), and the subsequent emergence of LLMs have established new performance benchmarks, showcasing remarkable capabilities in understanding nuanced text through prompting strategies like chain-of-thought (Wei et al., 2022).

Despite recent advances, state-of-the-art approaches are constrained by a uniform reasoning strategy ill-suited for the diverse nature of sarcasm. Different sarcastic expressions demand distinct analytical methods, such as contextual reasoning for expectation violations (Grice, 1975), cultural grounding for knowledge-dependent irony (Du et al., 2024), and pattern recognition for rhetorical devices (Gibbs Jr, 2000). This monolithic approach is further challenged by cognitive research indicating that human sarcasm processing involves parallel neural networks (Rapp et al., 2012), which suggests that single, unified models are inherently insufficient for this task. Consequently, even powerful LLMs lack the adaptivity to dynamically tailor their analysis, leading to inconsistent performance across varied types of sarcasm.

To overcome these limitations, we draw inspiration from the parallelism in human cognition to reframe sarcasm detection as a dynamic, adaptive reasoning process rather than a static classification task. We introduce RAM-SD, a Retrieval-Augmented Multi-Agent framework designed to

*Equal contribution.
†Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1434–1448

instantiate this vision. Central to our approach is a Meta-Planner that first analyzes the input text alongside retrieved contextual exemplars. Based on this analysis, it dynamically selects and dispatches a tailored ensemble of specialized agents to perform multi-faceted reasoning. The complementary analyses from these agents are then synthesized into a final, coherent prediction accompanied by an interpretable explanation.

This paper presents a comprehensive evaluation of RAM-SD, demonstrating its effectiveness and architectural integrity. Our primary contributions are as follows:

- We propose RAM-SD, a novel cognitively-inspired framework where a meta-planner orchestrates a retrieval-augmented multi-agent system to perform adaptive sarcasm analysis.
- The framework achieves state-of-the-art performance on four benchmarks, with an 77.74% average Macro-F1 score that outperforms GPT-4o+CoC by 7.01 points.
- It offers enhanced interpretability by generating structured reasoning traces that make the model's multi-stage analytical process transparent.

To the best of our knowledge, this is the first systematic instantiation of a retrieval-augmented, meta-planning multi-agent paradigm for sarcasm detection. The design improves interpretability via evidence-aligned rationales and reduces single-model bias through specialized agents and cross-checks, offering a principled pathway to reliable context-grounded reasoning and a promising direction for future work.

2 Related Work

2.1 Traditional Feature Engineering Approaches

Early computational approaches to sarcasm detection were characterized by their reliance on handcrafted features designed to capture explicit signals of irony. A seminal line of inquiry focused on the principle of incongruity, exemplified by the contrast hypothesis which identifies sarcasm through the co-occurrence of positive sentiment and a negative situation (Riloff et al., 2013). Following this, researchers broadened their investigation to include a diverse set of linguistic markers. These included lexical patterns such as specific n-grams and interjections (Davidov et al., 2010), overt typographical cues like exclamation marks and hashtags (Liebrecht et al., 2013), and various syntactic structures. The scope was further extended to incorporate pragmatic information, with studies demonstrating the utility of conversational context and author-specific attributes for more accurate classification (Bamman and Smith, 2015; Joshi et al., 2015). These methods, however, relied heavily on manual feature engineering and struggled to capture subtle sarcasm that required deep contextual or world knowledge, highlighting the need for architectures capable of learning complex representations directly from raw text.

2.2 The Deep Learning Revolution

The advent of deep learning instigated a paradigm shift from feature engineering to automated representation learning. This transition allowed models to learn salient features directly from raw text, significantly improving their ability to capture complex linguistic phenomena. Early applications in this domain included Recurrent Neural Networks (LSTMs), which proved effective for modeling the sequential dependencies inherent in conversational data (Ghosh et al., 2017), and Convolutional Neural Networks (CNNs), which were utilized to detect local, position-invariant patterns indicative of sarcasm (Misra and Arora, 2016). The field was fundamentally transformed by the arrival of pre-trained language models based on the Transformer architecture, such as BERT (Devlin et al., 2019). By leveraging deep contextualized embeddings, these models set new standards for performance and became the predominant approach for the task (Potamias et al., 2020). While these pre-trained models excelled at contextual understanding, their reliance on task-specific fine-tuning and their fixed architectures limited their adaptability, paving the way for more general, massively-scaled language models with emergent reasoning abilities.

2.3 Large Language Model Approaches

The current frontier in sarcasm detection is defined by the capabilities of LLMs such as the GPT series (Brown et al., 2020). Comprehensive evaluations have benchmarked these models, confirming their state-of-the-art performance while also revealing challenges related to prompt sensitivity and the comprehension of cultural nuances (Miranskyy et al., 2023; Zhang et al., 2024; Liu et al., 2025a). To elicit this performance, researchers

employ a range of strategies, from few-shot in-context learning to advanced prompting techniques like Chain-of-Thought, with recent work investigating the nature and efficacy of such step-by-step reasoning processes for sarcasm (Wei et al., 2022; Yao et al., 2025). While these methods enhance the reasoning of a single model, they do not alter its fundamental monolithic architecture. This uniform approach is inherently limited when faced with a cognitively complex task like sarcasm, which requires an orchestration of diverse analytical skills such as contextual reasoning and world knowledge (Grice, 1975; Rapp et al., 2012). Recent fine-grained and cross-dialectal evaluations further suggest that monolithic reasoning behaves unevenly across pragmatic sarcasm categories and English varieties. Motivated by this architectural mismatch, our work proposes a departure from the single-model paradigm towards a framework built on adaptive, specialized reasoning.

3 Methodology

We introduce **RAM-SD**, a Retrieval-Augmented Multi-Agent framework for Sarcasm Detection. As illustrated in Figure 1, RAM-SD operates through a four-stage cognitive pipeline: (1) Contextual Retrieval, (2) Retrieval-Augmented Meta-Planning, (3) Agent-based Multi-faceted Reasoning, and (4) Synthesis and Final Judgment.

3.1 Stage 1: Contextual Retrieval

This module grounds the analysis in relevant background knowledge. For a given query text T_q, we retrieve semantically relevant exemplars from a pre-compiled knowledge base $\mathcal{D} = \{(T_i, y_i)\}_{i=1}^{M}$.

Dual-Subset Retrieval. All texts are encoded into vectors using OpenAI's `text-embedding-3-large`. To mitigate label imbalance and enhance contrast, we partition the vectorized knowledge base into sarcastic $\mathcal{D}_{\text{vec}}^{\text{sarc}}$ and non-sarcastic $\mathcal{D}_{\text{vec}}^{\text{non}}$ subsets. We then retrieve the top-k exemplars from each subset based on similarity to T_q, forming a balanced retrieved set $\mathcal{C}_{\text{ret}} = \mathcal{C}_k^{\text{sarc}} \cup \mathcal{C}_k^{\text{non}}$ of size $2k$.

Rationale-Enhanced Contextualization. To transform retrieved instances into actionable insights, we use an LLM-based analyzer, LLM_{rat}, to generate a concise rationale r_i for each exemplar $(T_i, y_i) \in \mathcal{C}_{\text{ret}}$. The rationale explains *why* the text is sarcastic or not by identifying cues like hyperbole, sentiment-context mismatch, or knowledge contradictions. This creates a rationale-augmented context set, $\mathcal{C}_{\text{aug}} = \{(T_i, y_i, r_i)\}_{i=1}^{2k}$, which provides both examples and the underlying reasoning for downstream modules. These rationales are not used as a stand-alone classifier; instead, they function as a linguistically informed scaffold that exposes expectation violations, knowledge conflicts, and rhetorical cues for downstream reasoning.

3.2 Stage 2: Retrieval-Augmented Meta-Planning

This module acts as a dynamic planner, selecting a tailored reasoning strategy for the query T_q. The planner LLM_{plan} analyzes T_q by referencing the rationale-augmented exemplars in $\mathcal{C}_{\text{aug}}$, inferring potential context and background information. It produces two outputs: (1) the selected reasoning plan P^*, and (2) a contextual analysis O_{plan} that synthesizes inferred situational context based on $\mathcal{C}_{\text{aug}}$:

$$(P^*, O_{\text{plan}}) = \text{LLM}_{\text{plan}}(T_q, \mathcal{C}_{\text{aug}})$$

where $P^* = \arg\max_{P_i \in \mathcal{P}} P(P_i \mid T_q, \mathcal{C}_{\text{aug}})$ and O_{plan} contains inferred context that guides subsequent agent reasoning. Both P^* and O_{plan} are passed to Stage 3.

Pre-defined Reasoning Plans. We define three plans targeting distinct sarcasm archetypes:

- **Expectation Violation Plan ($\mathcal{P}_{\text{EV}}$):** Targets sarcasm arising from pragmatic incongruity. It deploys agents focused on semantics, expectations, incongruity, and rhetoric: $\{A_{\text{sem}}, A_{\text{exp}}, A_{\text{incon}}, A_{\text{rhet}}\}$.
- **Knowledge-Dependent Plan ($\mathcal{P}_{\text{KD}}$):** For sarcasm requiring external world knowledge. It uses agents for semantics, knowledge grounding, alignment, and rhetoric: $\{A_{\text{sem}}, A_{\text{know}}, A_{\text{align}}, A_{\text{rhet}}\}$.
- **Simple Irony Plan ($\mathcal{P}_{\text{SI}}$):** A lightweight plan for overt irony, primarily relying on the rhetoric agent $\{A_{\text{rhet}}\}$ for efficient detection, with optional $\{A_{\text{sem}}, A_{\text{incon}}\}$ support for ambiguous cases.

The output of this module is the selected plan P^*, which dictates the precise set of agents and the reasoning workflow for the next stage.

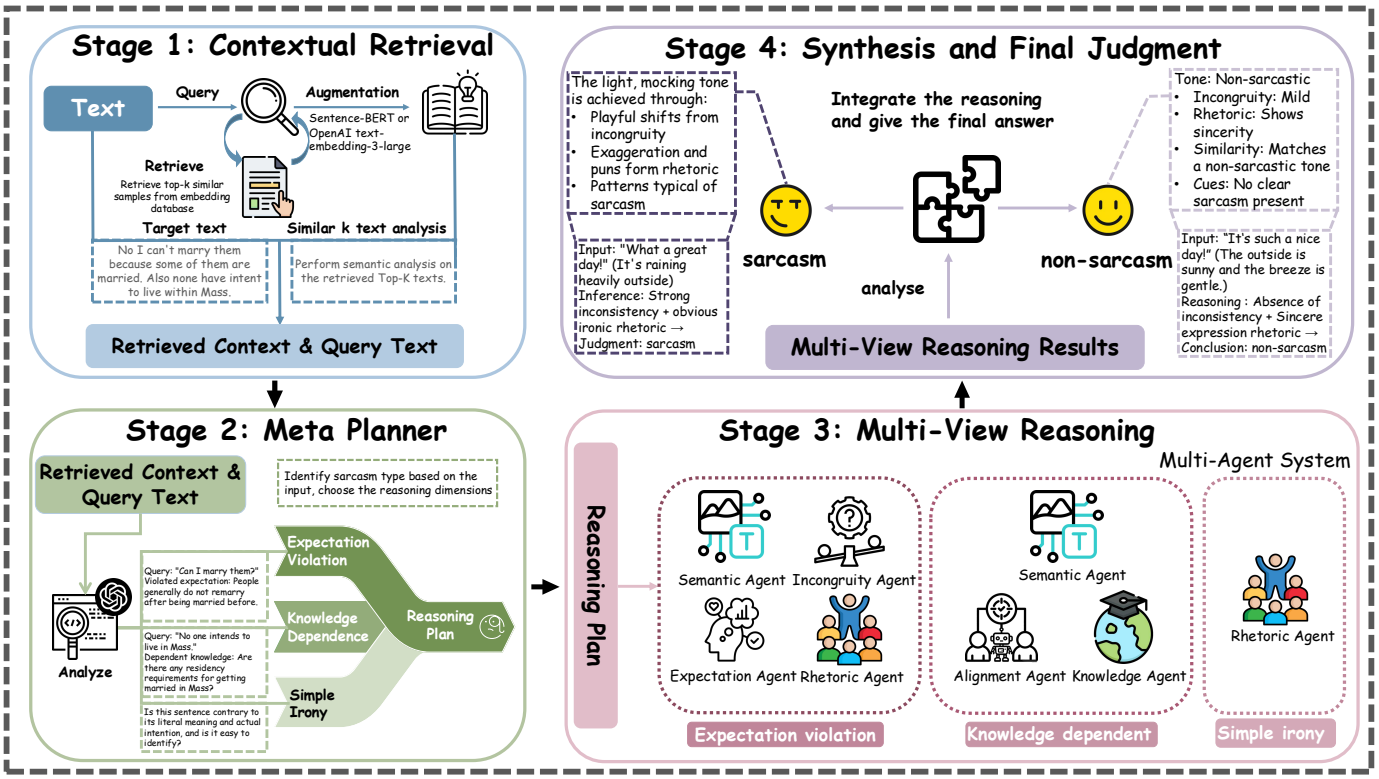

Figure 1: The four-stage architecture of the RAM-SD framework. Stage 1 grounds the query with retrieved sarcastic and non-sarcastic exemplars. This informs the Stage 2 meta-planner which selects a tailored reasoning plan. Stage 3 executes this plan with an ensemble of specialized agents. The agents' findings are synthesized in Stage 4 to produce a final judgment and explanation.

3.3 Stage 3: Agent-based Multi-View Reasoning

Following the meta-planner's directive, the selected agent ensemble P^* analyzes T_q from complementary perspectives. Each agent receives T_q, the contextual analysis O_{plan}, and the rationale-augmented exemplars $\mathcal{C}_{\text{aug}}$ as input. Agents operate through structured prompting, producing interpretable reasoning outputs:

- **Semantic Agent (A_{sem}):** Provides a literal interpretation of the text, identifying sentiment polarity and assessing surface-level coherence.
- **Rhetoric Agent (A_{rhet}):** Detects rhetorical signals such as hyperbole, understatement, or rhetorical questions, and qualitatively estimates their strength and intent through prompt-guided reasoning.
- **Expectation Agent (A_{exp}):** Examines the alignment between the situation implied by T_q and general world knowledge. Although not based on explicit statistical divergence, its reasoning approximates a measure conceptually akin to a deviation between expected and observed contexts.
- **Knowledge Agent (A_{know}):** Identifies key entities or events and retrieves relevant factual or cultural information through prompt-based grounding. The resulting summaries enrich contextual understanding.
- **Alignment Agent (A_{align}):** Evaluates whether the semantics of T_q are more consistent with sarcastic or non-sarcastic patterns in $\mathcal{C}_{\text{aug}}$, guided by O_{plan}.
- **Incongruity Agent (A_{incon}):** Synthesizes inconsistencies among semantic, rhetorical, and contextual dimensions into an overall incongruity judgment.

Together, these six agents form the minimal inventory we found sufficient to cover sarcasm's core mechanisms—expectation violation, knowledge conflict, and rhetorical contrast—while keeping the plan space compact. Each agent $A_j \in P^*$ produces a reasoning output $R_j = A_j(T_q, O_{\text{plan}}, \mathcal{C}_{\text{aug}})$. The collective outputs form a structured reasoning trace $\mathcal{R}_{P^*} = \{R_j \mid A_j \in P^*\}$, serving as the foundation for final synthesis.

3.4 Stage 4: Synthesis and Final Judgment

In the final stage, all agent outputs from $\mathcal{R}_{P^*}$ are first aggregated by an Integrator component, which synthesizes the diverse analytical perspectives into a coherent evidence summary. Subsequently, this integrated reasoning trace is evaluated by the Judger model, $\text{LLM}_{\text{judge}}$, to produce the final verdict. The Judger receives a comprehensive prompt $P_{\text{final}} = T_{\text{judge}}(T_q, \mathcal{R}_{P^*})$, which consolidates the original query T_q and the multi-faceted reasoning trace $\mathcal{R}_{P^*}$. This design ensures that the final prediction is grounded in the aggregated reasoning evidence while maintaining interpretability and consistency across agent decisions.

Prediction and Explanation. The Judger performs two crucial tasks. First, it predicts the final probability of the text being sarcastic:

$$p(y = \textit{sarcasm} \mid T_q, \mathcal{R}_{P^*}) = \text{LLM}_{\text{judge}}(P_{\text{final}})$$

The final label y_{pred} is determined by thresholding this probability at 0.5. Second, and critically for interpretability, it generates a natural-language explanation based on P_{final}, which is not a mere concatenation of agent outputs but a coherent synthesis that highlights the most salient evidence and logical steps leading to the conclusion. This dual output provides both a quantitative prediction and a qualitative justification, completing the RAM-SD pipeline by delivering a robust, transparent, and contextually aware sarcasm detection judgment.

4 Experiments

We conduct comprehensive experiments to evaluate the effectiveness of RAM-SD across diverse sarcasm detection benchmarks. Our evaluation focuses on: (1) Overall performance comparison with state-of-the-art baselines, (2) Ablation studies to validate the contribution of key architectural components, (3) Sensitivity analysis of the retrieval parameter, (4) Analysis of inference efficiency, and (5) Comprehensive evaluation of the framework's interpretability and limitations through quality assessment and error pattern analysis.

4.1 Experimental Setup

4.1.1 Datasets

We evaluate RAM-SD on four widely-used sarcasm detection benchmarks covering diverse linguistic styles and contexts. Table 1 summarizes the dataset statistics.

Table 1: Overview of the benchmark datasets used for evaluating sarcasm detection.

Dataset	Train	Test	Context	Source
IAC-V1	1,595	320	No	Reddit
IAC-V2	5,216	1042	No	Reddit
MUSTARD	552	138	Yes	TV Shows
SemEval	3,634	784	No	Twitter

IAC-V1/V2 (Oraby et al., 2016) contain political debate posts from Reddit's *r/politics* forum, where sarcasm often manifests through expectation violations and implicit critique. **MUSTARD** (Ghosh et al., 2020) provides conversational utterances from TV show dialogues with contextual background, requiring understanding of character relationships and situational awareness. **SemEval-2018 Task 3** (Van Hee et al., 2018) comprises Twitter posts where sarcasm frequently relies on simple irony and rhetorical devices within character-limited expressions.

4.1.2 Baselines

We compare RAM-SD against representative methods from three categories:

Deep Learning Models: **MIARN** (Tay et al., 2018) employs multi-interactive attention networks; **SAWS** (Pan et al., 2020) leverages sentiment-aware word selection; **DC-Net** (Liu et al., 2021) utilizes dual-channel networks for multi-view reasoning.

Fine-tuned PLMs: **BERT-base** (Devlin et al., 2019) and **RoBERTa-base** (Liu et al., 2019) fine-tuned on each target dataset using standard classification protocols.

LLM-based Methods: We compare RAM-SD against several state-of-the-art zero-shot LLM baselines. Our primary comparisons include prompting strategies from SarcasmCue (Yao et al., 2025), such as GPT-4o with Chain of Contradiction, Graph of Cues, and Bagging of Cues prompting, as well as **IDADP** (Yi et al., 2025) and **CAF-I** (Liu et al., 2025b). Additional open-source reproducibility results are reported in Appendix A.

4.1.3 Implementation Details

We use OpenAI's `text-embedding-3-large` (3072 dimensions) for text embeddings in the Contextual Retrieval stage. The Meta-Planner, all specialized agents, and the Judger are implemented using GPT-4o with temperature=0.1, top-p=1.0, max_tokens=512, and default penalties

(`frequency_penalty=0`, `presence_penalty=0`). We use a single generation per prompt (`n=1`). Because the API does not expose seed control and we do not use stochastic resampling, each input yields one stable output. Vector similarity search employs FAISS (Johnson et al., 2019) with cosine similarity after L2 normalization. Retrieval is performed strictly over the training split of the same dataset, with no cross-dataset retrieval and no test-set access. Unless otherwise specified, we set $k = 3$ for dual-subset retrieval, yielding $|\mathcal{C}_{\text{ret}}| = 6$ balanced exemplars. The $\pm$ values in Tables 2 and 3 reflect three runs differing only in FAISS retrieval ordering, and all p-values are computed using paired McNemar tests against GPT-4o+CoC.

4.2 Main Results

Table 2 presents the performance comparison across all datasets. RAM-SD achieves state-of-the-art performance with an average Macro-F1 of 77.74% and accuracy of 77.65%, substantially outperforming all baseline methods.

Performance Analysis: RAM-SD achieves consistent improvements across all datasets, with particularly strong performance on SemEval (86.30% Accuracy) where retrieval-grounded reasoning is especially beneficial. The system outperforms the strongest baseline GPT-4o+CoC by 7.01 points on average (70.73% vs. 77.74%). On MUSTARD, we intentionally disable Stage 1 retrieval because the dataset already provides rich dialogue context and its training instances do not naturally serve as independent retrieval exemplars; under this setting, RAM-SD is therefore equivalent to RAM-SD w/o RAG. Appendix A further reports an open-source reimplementation with DeepSeek-V3 on IAC-V1, where RAM-SD preserves a 3.31-point Ma-F1 advantage over the direct baseline.

4.3 Ablation Study

To validate the architectural design of RAM-SD, we conduct ablation studies isolating the contribution of key components. Table 3 presents results averaged across all four datasets.

The most substantial performance degradation arises from removing the **Meta-Planner (Stage 2)**, which causes a **3.27%** Ma-F1 drop. This highlights that adaptive reasoning-plan selection is crucial for handling the diverse manifestations of sarcasm. The **Incongruity Agent** follows with a **3.09%** decrease, underscoring the necessity of detecting semantic-pragmatic inconsistencies. Eliminating the **Contextual Retrieval** leads to a **2.69%** reduction, confirming that grounding analysis in contextual exemplars remains vital for recognizing nuanced sarcastic patterns. The **Rhetoric Agent** yields a **2.57%** decline, showing that rhetorical-cue modeling provides complementary interpretive depth. Finally, removing the **Knowledge Agent** results in a **2.43%** drop, suggesting that world knowledge enhances contextual understanding without dominating the reasoning process.

4.4 Retrieval Parameter Sensitivity

We analyze the sensitivity of RAM-SD to the retrieval parameter k, which controls the number of exemplars retrieved from each subset (sarcastic and non-sarcastic). Figure 2 shows Macro-F1 performance on IAC-V1 as k varies from 1 to 10.

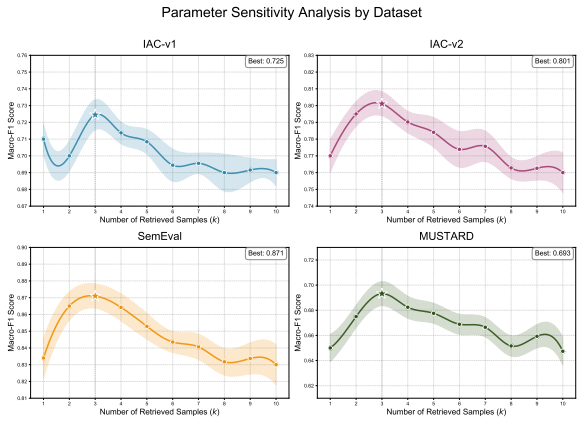

Figure 2: Retrieval sensitivity across the four datasets. Performance peaks at $k = 3$ and then plateaus.

Performance improves as k increases from 1 to 3, confirming that richer contextual grounding benefits downstream reasoning. However, performance plateaus at $k = 3$ and slightly declines for $k > 5$, suggesting that excessive exemplars introduce noise from less relevant samples. This validates our default choice of $k = 3$ as balancing contextual richness with focused relevance. To probe robustness under weak retrieval signals relevant to rarely attested but still observed cases, we additionally replace two of the three retrieved exemplars with random training samples for every IAC-V1 test instance. Under this severe stress test, Ma-F1 drops only from 74.45 to 72.71 (-1.74), indicating graceful rather than catastrophic degradation; details are provided in Appendix C.

Table 2: Overall performance comparison across four benchmark datasets. All LLM strategies are zero-shot. Acc. denotes Accuracy and Ma-F1 signifies Macro-F1. Best results are in bold, second-best are underlined. For RAM-SD, $\pm$ values reflect three runs differing only in FAISS retrieval ordering; p-values are paired McNemar tests against GPT-4o+CoC.

Method	**IAC-V1**		**IAC-V2**		**MUSTARD**		**SemEval 2018**		**Avg.**	
	Acc.	Ma-F1	Acc.	Ma-F1	Acc.	Ma-F1	Acc.	Ma-F1	Acc.	Ma-F1
MIARN	63.21	63.18	72.75	72.75	64.60	63.90	68.50	67.80	67.26	66.91
SAWS	66.13	65.60	76.20	76.20	69.71	70.95	69.90	68.90	70.48	70.41
DC-Net	66.50	66.40	78.00	77.90	71.28	71.43	76.30	76.70	73.02	73.11
BERT	65.30	65.20	76.40	76.20	64.30	64.30	69.90	68.40	68.97	68.52
RoBERTa	70.10	69.90	76.60	76.70	66.10	66.00	70.20	69.10	70.75	70.42
GPT-4o	70.63	70.05	73.03	71.99	67.24	65.79	64.03	63.17	68.73	67.75
GPT-4o+CoT	61.56	58.49	58.83	56.42	58.92	51.99	58.11	55.76	59.36	55.67
GPT-4o+CoC	72.19	71.52	73.36	72.31	69.42	68.48	70.79	70.60	71.44	70.73
GPT-4o+GoC	69.84	69.20	72.47	71.60	68.10	67.42	65.91	65.03	69.58	68.31
GPT-4o+BoC	70.32	70.11	73.12	72.08	67.53	66.12	64.88	63.92	68.96	68.06
CAF-I	73.75	73.71	77.80	76.82	**75.21**	**74.73**	80.73	79.99	76.89	76.31
IDADP	65.84	67.13	70.32	69.73	67.44	67.37	65.31	65.28	67.22	67.37
RAM-SD w/o RAG	71.25	71.23	78.69	78.68	69.38	69.32	80.94	80.90	75.07	75.05
RAM-SD	**74.81**	**74.45**	**80.13**	**80.11**	69.38	69.32	**86.30**	**87.10**	**77.65**	**77.74**
Improv.	1.06↑	0.74↑	2.13↑	2.21↑	-	-	5.36↑	6.20↑	0.27↑	0.94↑
p-val.	$5.82e^{-3}$	$1.91e^{-4}$	$7.04e^{-3}$	$3.36e^{-3}$	$9.15e^{-4}$	$2.28e^{-3}$	$6.77e^{-3}$	$4.59e^{-4}$	$8.13e^{-3}$	$3.51e^{-3}$

Table 3: Ablation Study Results (Average Macro-F1% across Four Datasets)

Configuration	Ma-F1	ΔMa-F1
Full RAM-SD System	**77.74** $\pm$ 0.8	-
w/o Contextual Retrieval (Stage 1)	75.05 $\pm$ 0.9	-2.69
w/o Meta-Planner (Stage 2)	74.47 $\pm$ 1.1	-3.27
w/o Rhetoric Agent	75.17 $\pm$ 0.7	-2.57
w/o Knowledge Agent	75.31 $\pm$ 0.8	-2.43
w/o Incongruity Agent	74.65 $\pm$ 0.9	-3.09

4.5 Inference Efficiency Analysis

Beyond accuracy improvements, the meta-planner provides computational efficiency through adaptive agent selection. Table 4 presents inference time breakdown across the four-stage pipeline.

Table 4: Average Inference Time per Sample

Stage	Time (s)	Percentage
Stage 1: Contextual Retrieval	3.42	17.67%
Stage 2: Meta-Planning	2.41	12.46%
Stage 3: Multi-Agent Reasoning	11.35	58.65%
Stage 4: Synthesis & Judgment	2.17	11.21%
Total	**19.35**	**100%**

Multi-agent reasoning dominates inference time at 58.67%, but the meta-planner mitigates this by selecting minimal agent ensembles. Simple Irony plans activate only 1–3 agents compared to 4–5 for Expectation Violation plans, achieving 35% reduction in agent invocations for appropriate cases. The reported 19.35s is a sequential upper bound rather than a deployment-time constant: the three plans require only 4, 6, and 7 API calls for Simple Irony, Expectation Violation, and Knowledge-Dependent reasoning, respectively, and the agent calls can be executed asynchronously. In practice, with parallel execution, processing 320 samples completes within approximately 20 minutes (Appendix C).

4.6 Interpretability Quality Evaluation

To validate that RAM-SD produces higher-quality explanations beyond accuracy gains, we conduct **LLM-as-Judge** evaluation on 200 randomly sampled instances (50 per dataset, balanced) comparing four systems: **RAM-SD**, **RAM-SD w/o RAG**, **GPT-4o+CoC**, and **GPT-4o**. Using GPT-4o as the primary judge, we evaluate explanations along four criteria: **Correctness** (faithfulness to the text and agent evidence), **Sufficiency** (coverage of major reasoning elements), **Coherence** (logical consistency across the reasoning chain), and **Specificity** (reference to concrete linguistic or factual cues). Appendix B further reports a blind manual evaluation with two trained annotators and cross-judge validation with Gemini-2.5 Flash.

Results. As shown in Table 5, RAM-SD achieves an overall score of 4.43, representing a significant improvement of 16.5% over GPT-4o+CoC (score of 3.80) and 38.5% over the standard GPT-4o (score of 3.20). Notably, while GPT-4o+CoC demonstrates highly competitive performance in **Specificity** with a score of 4.26 that approaches

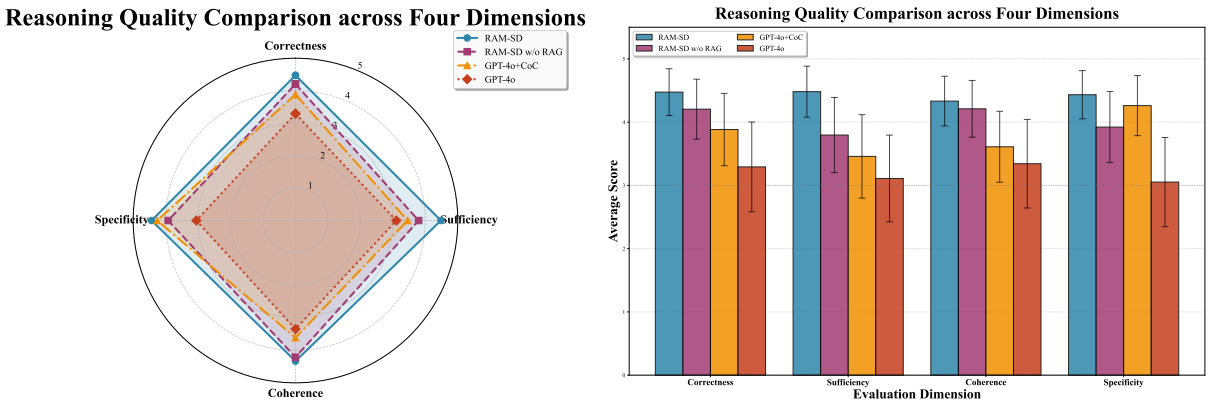

Figure 3: LLM-as-Judge evaluation results. **Left:** Radar chart across four dimensions, RAM-SD (blue) dominates, while GPT-4o+CoC (orange) achieves competitive Specificity, surpassing RAM-SD w/o RAG (purple). **Right:** Grouped bar chart showing mean scores with standard deviation bars for each dimension.

Table 5: GPT-4o-as-Judge evaluation on $n = 200$ instances. Scores are reported on a 5-point Likert scale ($5 = best$) for Correctness (Corr.), Sufficiency (Suff.), Coherence (Cohe.), and Specificity (Spec.). All improvements of RAM-SD over the baselines are statistically significant.

System	Corr.	Suff.	Cohe.	Spec.	Overall
RAM-SD	**4.47**	**4.48**	**4.33**	**4.43**	**4.43**
RAM-SD w/o RAG	4.20	3.80	4.21	3.92	4.03
GPT-4o+CoC	3.88	3.46	3.61	4.26	3.80
GPT-4o	3.29	3.11	3.34	3.05	3.20

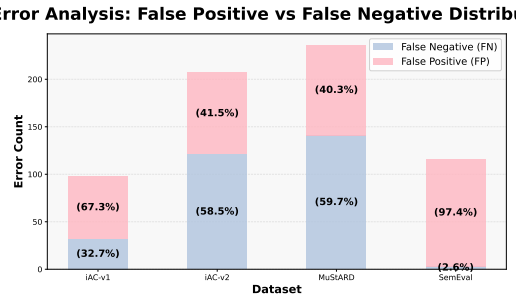

Figure 4: Error distribution showing distinct FP/FN patterns across datasets.

RAM-SD's 4.43, the advantages of our framework are most pronounced in **Sufficiency** and **Coherence**. In these dimensions, RAM-SD outperforms GPT-4o+CoC by 29.6% and 20.0% respectively, which we attribute to the richer evidence provided by the multi-agent architecture and the consistency enforced by the integrator. The ablation result for RAM-SD w/o RAG, which scored 4.03, confirms that the retrieval mechanism contributes a substantial 0.40 points (a 9.0% relative contribution) to the final performance. This ranking is stable under human and cross-judge validation: Appendix B shows a blind manual study with two annotators (4.39 vs. 4.05 overall for RAM-SD vs. GPT-4o+CoC) and a Gemini-2.5 Flash re-evaluation on the same 200 instances (4.34 vs. 3.82).

Figure 3 visualizes these results. The radar chart reveals an interesting pattern: while RAM-SD dominates most dimensions, GPT-4o+CoC achieves competitive Specificity scores, indicating calibration prompting's strength in producing concrete reasoning. The grouped bar chart with error bars clearly shows dimension-wise performance differences and variability across systems.

Implications. RAM-SD's structured reasoning enables transparent error diagnosis: users can inspect (1) retrieved exemplar relevance, (2) meta-planner strategy selection, (3) individual agent contributions, and (4) integrator synthesis. This interpretability is crucial for high-stakes deployments requiring human oversight. Appendix D extends this discussion with an additional knowledge-dependent success case in which direct prompting over-predicts sarcasm but RAM-SD correctly identifies sincere argumentative discourse.

4.7 Error Analysis and Failure Patterns

4.7.1 Error Analysis

Figure 4 shows error distributions across datasets, revealing retrieval-induced failure modes.

Corpus Size Effects: Our error analysis reveals a strong correlation between corpus characteristics and predominant error patterns, creating a dichotomy between datasets prone to over-prediction and those susceptible to under-detection. The IAC-v1 and SemEval datasets exhibit high false positive (FP) rates of 67.3% and 97.4%, respectively.

This over-prediction in IAC-v1 is attributed to its small retrieval pool (1,595 samples), which forces a reliance on surface-level rhetorical markers that can misclassify figurative language. SemEval's FP bias stems from misleading structural cues, such as hashtags and @mentions, that superficially mimic sarcasm without the same pragmatic intent.

Conversely, the IAC-v2 and MUSTARD datasets show significant under-detection, with false negative (FN) rates of 58.5% and 59.7%. The larger corpus of IAC-v2 appears to induce a more conservative model behavior, as the abundance of diverse retrieved contexts dilutes confidence in more nuanced cases. Finally, MUSTARD's FN dominance highlights the inherent limitation of a text-only analysis on multimodal data, where essential audio-visual cues like vocal tone and facial expressions are fundamentally inaccessible.

4.7.2 Case Study: Cascading Failure from Retrieval Bias

We examine a representative SemEval FP error: *"when you refer to yourself in the plural, you'll get that @RBRNetwork1..."* (Label: 0, Predicted: 1, Conf.: 0.85). Table 6 traces the cascading failure.

Table 6: Stage-wise error propagation analysis.

Stage	Output & Failure Mechanism
Retrieval	Retrieved: *"LOOL from guy with multiple handles"*, *"9 Followers, now relevant?"* **Issue:** Matched Twitter structure (@mentions), not pragmatic intent (mockery vs. meta-commentary).
Meta-Plan	Plan: $\mathcal{P}_{\text{EV}}$; O_{plan}: "indirect mockery" **Issue:** Biased priming from retrieval.
Agents	A_{rhet}: "mild irony"; A_{incon}: 6/10; A_{exp}: "mocks plural self-ref" **Issue:** Weak signals amplified by biased O_{plan}.
Synthesis	Judger: $p = 0.85$; "strong sarcastic alignment" **Issue:** No mechanism to challenge consensus.

This exemplifies the *echo chamber effect*: biased retrieval constrains meta-planning, which primes agents toward confirmation, yielding high-confidence errors. Our symbolic pipeline lacks gradient-based error attribution, making such cascades hard to detect. Mitigation requires *contrastive retrieval* and *agent-level calibration* to challenge weak unanimous signals.

5 Conclusion

We introduced RAM-SD, a novel four-stage multi-agent framework for sarcasm detection. Its core meta-planner dynamically orchestrates specialized agents based on retrieval-augmented analysis of the input text. The framework achieves a state-of-the-art 77.74% average Macro-F1, improving upon the GPT-4o+CoC baseline by 7.01 points. Ablation studies validate this hierarchical design, confirming the meta-planner and retrieval module are critical components that contribute 3.27 and 2.69 percent to the Macro-F1 score respectively. Our results demonstrate the efficacy of structured multi-agent systems for nuanced language understanding. We position RAM-SD as an extensible, backward-compatible paradigm: practitioners can plug in new plans/agents as domains evolve. We expect this line of work to guide a shift from opaque single-pass classifiers to context-grounded, plan-conditioned reasoning frameworks.

Limitations

Fundamental Limitations of Retrieval-Based Reasoning. Our primary limitation concerns entirely novel sarcasm forms rather than rare but still attested cases. When relevant precedents exist but retrieval signals are weak, RAM-SD degrades gracefully: even under a severe corruption setting where two of the three retrieved exemplars are replaced with random samples, performance drops by only 1.74 Ma-F1 on IAC-V1 (Appendix C). The harder failure mode arises when no meaningful precedent exists in the retrieval corpus at all, as in emergent memes, evolving cultural references, or multimodal sarcasm such as MUSTARD where text alone cannot capture audio-visual cues. In such cases, retrieval alone is insufficient, and stronger compositional reasoning or multimodal world modeling is needed for genuine generalization.

Agent Coordination and Conflict Resolution. While the Integrator synthesizes agent outputs, our framework lacks principled mechanisms for contradictory evidence. Current reliance on implicit weighting by the Judger is brittle. Future work should incorporate confidence calibration, cross-agent consistency checks, or argumentation-theoretic arbitration protocols.

Acknowledgments

This research was supported by National Natural Science Foundation of China (Grant No. 62502396), XJTLU Research Development Fund (RDF-24-02-008), and Suzhou Industrial Park Interdisciplinary Innovation Research Platform for Affective Computing and Interactive Health (CXK 2025101).

References

David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. In *Ninth international AAAI conference on web and social media*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics.

Yiran Du, Huimin He, and Zihan Chu. 2024. Cross-cultural nuances in sarcasm comprehension: a comparative study of chinese and american perspectives. *Frontiers in Psychology*, 15:1349002.

Aniruddha Ghosh, Debanjan Ghosh, and Smaranda Muresan. 2020. MUSTARD: A multimodal sarcasm detection framework. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4078–4089. Association for Computational Linguistics.

Debanjan Ghosh, Alexander R Fabbri, and Smaranda Muresan. 2017. The role of conversation context for sarcasm detection in online interactions. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 186–196. Association for Computational Linguistics.

Raymond W Gibbs Jr. 2000. Irony in talk among friends. *Metaphor and Symbol*, 15(1-2):5–27.

Herbert Paul Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys*, 50(5):1–22.

Aditya Joshi, Vaibhav Sharma, and Pushpak Bhattacharyya. 2015. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers*, pages 2482–2491.

Christine Liebrecht, Florian Kunneman, and Antal van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets # not. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 29–37. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yiyi Liu, Yequan Wang, Aixin Sun, Xuying Meng, Jing Li, and Jiafeng Guo. 2021. A dual-channel framework for sarcasm recognition by detecting sentiment conflict. *arXiv preprint arXiv:2109.03587*.

Ziqi Liu, Yangbin Chen, Ziyang Zhou, Yilin Li, Mingxuan Hu, Yushan Pan, and Zhijie Xu. 2025a. Sevade: Self-evolving multi-agent analysis with decoupled evaluation for hallucination-resistant irony detection. *arXiv preprint arXiv:2508.06803*.

Ziqi Liu, Ziyang Zhou, and Mingxuan Hu. 2025b. Caf-i: A collaborative multi-agent framework for enhanced irony detection with large language models. *arXiv preprint arXiv:2506.08430*.

Andriy Miranskyy, Mehdi Hooshangi, Mohammad Tawalbeh, and Fadi Moussa. 2023. On sarcasm detection with OpenAI GPT-based models. *arXiv preprint arXiv:2312.04642*.

Rishabh Misra and Prahal Arora. 2016. Using multilayer perceptrons for multimodal sarcasm detection. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 32–39.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. 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*, pages 31–41. Association for Computational Linguistics.

Hongru Pan, Zhiwei Lin, Peng Fu, and Weihao Wang. 2020. Modeling sentiment-aware word selection for multimodal sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2378–2388. Association for Computational Linguistics.

Rolandos Alexandros Potamias, Georgios Siolas, and Andreas-Georgios Stafylopatis. 2020. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 32(23):17309–17320.

Alexander M Rapp, Diana E Mutschler, Barbara Wild, Michael Erb, Inge Lengsfeld, Dorothee Saur, and Wolfgang Grodd. 2012. Lateral inferences and scalar implicatures. *NeuroImage*, 59(4):3103–3113.

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

Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856*.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 Task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Ben Yao, Yazhou Zhang, Qiuchi Li, and Jing Qin. 2025. Is sarcasm detection a step-by-step reasoning process in large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25651–25659.

Peiling Yi, Yuhan Xia, and Yunfei Long. 2025. Irony detection, reasoning and understanding in zero-shot learning. *IEEE Transactions on Artificial Intelligence*.

Yazhou Zhang, Chunwang Zou, Zheng Lian, Prayag Tiwari, and Jing Qin. 2024. SarcasmBench: Towards evaluating large language models on sarcasm understanding. *arXiv preprint arXiv:2408.11319*.

A Additional Experimental Details

This appendix summarizes reproducibility clarifications and supplementary experiments omitted from the main text for space reasons.

A.1 Reproducibility and Retrieval Protocol

We use a deterministic decoding configuration for all GPT-4o calls: temperature=0.1, top-p=1.0, $n = 1$, frequency_penalty=0, and presence_penalty=0. The API does not expose seed control, and we do not use stochastic resampling. Accordingly, repeated runs differ only in FAISS retrieval ordering rather than in LLM generation.

Retrieval follows a strict train-only protocol: each dataset retrieves exemplars only from its own training split; we never retrieve from the test split and never retrieve across datasets. Labels on retrieved exemplars come exclusively from the training data and are used only as in-context demonstrations. For MUSTARD, we disable Stage 1 retrieval because the dataset already provides rich dialogue context and its training instances do not naturally serve as independent retrieval exemplars; under this setting, RAM-SD is therefore equivalent to RAM-SD w/o RAG.

Table 7: Leakage stress test on IAC-V1.

Configuration	Ma-F1	Observation
Train-only	74.45	No leakage
10% test added	81.23	+6.78 inflation

A.2 Open-Source Reproducibility

To verify that the gains are not tied to a proprietary model stack, we reimplemented RAM-SD with DeepSeek-V3 on IAC-V1.

Table 8: Open-source reproducibility results on IAC-V1.

Model	Accuracy	Ma-F1
RAM-SD (DeepSeek-V3)	72.81	72.79
DeepSeek-V3 baseline	70.31	69.48

This preserves gains of 2.50 Accuracy points and 3.31 Ma-F1 points over the direct DeepSeek-V3 baseline, showing that the framework remains beneficial in an open-source setting.

B Explanation Evaluation Protocol and Validation

We supplement the GPT-4o-as-judge results with explicit criteria, a blind manual study, and cross-judge validation.

Table 9: Explanation evaluation criteria.

Criterion	Definition
Correctness	Explanation faithfully reflects the input text and agent evidence without hallucination.
Sufficiency	All major reasoning elements, such as expectation violations, rhetorical cues, and relevant knowledge, are covered.
Coherence	The reasoning chain is logically consistent and integrates multi-agent evidence without contradiction.
Specificity	The explanation cites concrete cues, such as lexical contrast or factual inconsistency, rather than generic statements.

B.1 Manual Evaluation

Setup: two annotators with linguistics/pragmatics training rated 200 paired explanations under blind 1–5 Likert scoring; disagreements were adjudicated through discussion.

Table 10: Blind manual evaluation on 200 paired explanations.

System	Corr.	Suff.	Cohe.	Spec.	Overall
RAM-SD	4.38	4.45	4.41	4.32	4.39
GPT-4o+CoC	4.06	4.04	4.04	4.04	4.05
Δ	+0.32	+0.41	+0.37	+0.28	+0.34

Manual annotators consistently preferred RAM-SD when the explanation integrated multiple evidence types rather than relying on a single sentiment-like cue.

B.2 Cross-Judge Validation

We further re-evaluated the same 200 instances with Gemini-2.5 Flash as an independent judge.

Table 11: Cross-judge validation with Gemini-2.5 Flash on the same 200 instances.

System	Corr.	Suff.	Cohe.	Spec.	Overall
RAM-SD	4.35	4.32	4.38	4.31	4.34
GPT-4o+CoC	3.82	3.78	3.85	3.83	3.82

The same ranking is preserved across human and independent-model judges, which suggests that the main-text explanation gains are not an artifact of a single evaluator.

C Additional Robustness and Generalization Studies

C.1 Weak-Retrieval Robustness

To simulate rarely attested but still observed cases, we replaced two of the three retrieved exemplars with random training instances for every IAC-V1 test sample.

Table 12: Robustness under corrupted retrieval on IAC-V1.

Retrieval configuration	Ma-F1	Degradation
Normal (top-3)	74.45	–
67% corrupted (2/3 random)	72.71	-1.74

Even under this severe upper-bound stress test, performance degrades gracefully rather than catastrophically, indicating that Stage 1 provides approximate guidance instead of deterministic rules.

C.2 Stage-1 Rationale Verification

To assess whether rationale-augmented retrieval captures human-interpretable evidence, two annotators judged 100 Stage 1 rationale chains.

Table 13: Human verification of Stage 1 rationales.

Outcome	Ratio
Correct	74%
Partially correct	21%
Incorrect	5%

We also observed that expert agreement with the model's routing increased from 72% to 84% after annotators reviewed the Stage 1 rationales, suggesting that the rationales make implicit pragmatic cues more explicit.

C.3 API Call Complexity

To provide a hardware-independent cost view, we report API-call counts for each reasoning plan.

Table 14: API calls per reasoning plan.

Plan type	API calls
Simple Irony	4
Expectation Violation	6
Knowledge-Dependent	7

C.4 Cross-Dialectal Generalization

We further evaluated RAM-SD on BESSTIE dialectal English subsets.

Table 15: Cross-dialectal evaluation on BESSTIE.

Dialect	RAM-SD	GPT-4o	SOTA	vs. SOTA	vs. GPT-4o
en-IN	83.51	74.95	58.00	+25.51	+8.56
en-UK	83.18	82.05	77.00	+6.18	+1.13
en-AU	81.40	78.98	72.00	+9.40	+2.42

The strongest gain appears on Indian English, suggesting that retrieval-grounded context and adaptive plan selection are especially helpful when sarcasm depends on culturally grounded knowledge.

D Plan-Aligned Case Analysis

This appendix complements the main-text SemEval failure case with an additional success case from the knowledge-dependent regime.

D.1 Knowledge-Dependent Success Case

Text. "Yes it is, police aren't obligated by the US Constitution to protect individual citizens. If they were, lawsuits could be filed every time a crime is committed, and the police would go bankrupt really quick."

Ground truth / prediction. Ground-truth label 0 (non-sarcastic); RAM-SD predicts 0 with

confidence 0.95. Direct GPT-4o prompting and CoT-style baselines tend to over-predict sarcasm here because they misread argumentative hyperbole ("go bankrupt really quick") as ironic exaggeration.

Why RAM-SD succeeds. Stage 1 retrieves non-sarcastic legal and policy arguments whose rationales emphasize literal reasoning rather than mockery. The meta-planner therefore selects the knowledge-dependent plan. The Knowledge Agent grounds the claim in legal precedent and institutional constraints, the Alignment Agent finds that the literal meaning is consistent with common legal knowledge, and the Semantic Agent identifies a literal argumentative tone rather than semantic inversion. The final judgement is thus driven by agreement between retrieval, planning, and agent-level evidence, showing how RAM-SD can suppress false positives when emphatic language appears inside sincere argumentative discourse.

E Full Prompt Templates

This appendix provides the complete prompt templates used in each stage of the RAM-SD framework. All prompts are executed using GPT-4o with temperature=0.1 for consistency.

E.1 Stage 1: Contextual Retrieval

E.1.1 Step 1.1: Rationale Generation Prompt

This prompt generates explanatory rationales for each retrieved exemplar to create rationale-augmented context $\mathcal{C}_{\text{aug}}$.

Rationale Generator

```
Analyze why the following text is [sarcastic/non-sarcastic]:
Text: "{text}" Label: {label}
Provide a concise rationale (2-3 sentences) explaining: 1.
Key linguistic features (tone, style, rhetorical devices) 2.
Contextual cues or knowledge dependencies 3. Why this is
clearly [sarcastic/non-sarcastic]
Focus on concrete evidence rather than abstract concepts.
```

Note: This prompt is applied to all $2k$ retrieved exemplars to form the rationale-augmented context set $\mathcal{C}_{\text{aug}} = \{(T_i, y_i, r_i)\}$.

E.1.2 Step 1.2: Similarity Analysis Prompt

After generating rationales, this prompt analyzes the query's similarity patterns with both sarcastic and non-sarcastic exemplars. The analysis output informs the Meta-Planner in Stage 2.

Similarity Analyzer

```
You are a similarity analysis expert. Compare the query text
with retrieved exemplars to identify linguistic patterns and
stylistic similarities.
**Query Text:** "{text}" {context_if_available}
**Retrieved NON-SARCASTIC Examples (Label 0) with
Rationales:** 1. Text: "{example_1}" | Rationale:
{rationale_1} 2. Text: "{example_2}" | Rationale:
{rationale_2} 3. Text: "{example_3}" | Rationale:
{rationale_3}
**Retrieved SARCASTIC Examples (Label 1) with Rationales:**
1. Text: "{example_1}" | Rationale: {rationale_1} 2.
Text: "{example_2}" | Rationale: {rationale_2} 3. Text:
"{example_3}" | Rationale: {rationale_3}
Analyze the query's similarity patterns:
**Similarity to NON-SARCASTIC Examples:** - Shared features:
[tone, directness, literal expression, etc.] - Pattern
alignment: [similar linguistic structures or rhetorical
styles] - Strength of similarity: [strong/moderate/weak]
**Similarity to SARCASTIC Examples:** - Shared features:
[irony markers, contradiction patterns, mocking tone, etc.]
- Pattern alignment: [similar ironic devices or expectation
violations] - Strength of similarity: [strong/moderate/weak]
**Comparative Assessment:** - Primary similarity direction:
[more similar to sarcastic/non-sarcastic/mixed] - Key
discriminative features: [2-3 critical differences from one
category] - Confidence level: [high/medium/low]
**Contextual Inference:** Based on retrieved examples and
their rationales, infer: - Likely situational context
of the query - Potential background knowledge required
for interpretation - Candidate sarcasm type if sarcastic:
[expectation_violation/knowledge_dependent/ simple_irony]
Return structured analysis focusing on concrete pattern
matching with exemplars.
```

Output: This produces similarity analysis S_{analysis} that captures pattern-matching insights between query and exemplars, which is passed to Meta-Planner along with $\mathcal{C}_{\text{aug}}$.

E.2 Stage 2: Retrieval-Augmented Meta-Planning

E.2.1 Meta-Planner Prompt

This prompt integrates similarity analysis from Stage 1 with feature analysis to select optimal reasoning plan. Produces both plan selection (P^*) and contextual analysis (O_{plan}).

Meta-Planner (Retrieval-Augmented)

```
As meta-planner, analyze query and select optimal reasoning strategy.
**Query:** "{text}"
**Similarity Analysis from Stage 1:** {similarity_analysis_output}
**Retrieved Examples Summary:** - Non-sarcastic:
{brief_summary_of_non_sarc_examples} - Sarcastic:
{brief_summary_of_sarc_examples}
Based on similarity patterns and query features, analyze:
**Feature Analysis:** 1. Contradiction level: [none/low/medium/high]
- between literal/intended meaning 2. Exaggeration/irony present:
[yes/no] 3. Emotional conflict: [yes/no] - positive words with negative
intent 4. Context dependency: [low/medium/high] 5. Rhetorical devices:
[list key devices if present] 6. Knowledge entities: [list SPECIFIC
entities: named persons, events, orgs] Exclude: generic concepts,
common terms, abstract ideas
**Plan Selection (choose ONE):** - expectation_violation: For
contradiction/irony, pragmatic violations - knowledge_dependent: For
SPECIFIC entities requiring background knowledge - simple_irony: For
short (fewer than 15 words), overt ironic expressions
**Contextual Analysis (O_plan):** Synthesize inferred context from
similarity analysis and examples: - Likely situational context:
[inferred scenario/setting] - Relevant background knowledge: [key
info for understanding] - Expectation baseline: [what would be normal
expression in this context] - Pragmatic interpretation hints: [guidance
for downstream agents]
**Output:** {"selected_plan":
"expectation_violation/knowledge_dependent/simple_irony",
"confidence": 0-1, "contextual_analysis": "synthesized context and
interpretation guidance", "reasoning": "why this plan based on
similarity patterns and features"}
Priority: Weight similarity patterns heavily - if query strongly
matches sarcastic/ non-sarcastic exemplar patterns, factor this into
plan selection.
```

Output: Produces (P^*, O_{plan}) where P^* determines agent ensemble for Stage 3, and O_{plan} pro-

vides contextual grounding for all agents.

Implementation Note: LLM selection validated by rules (text length greater than 50 words overrides `simple_irony`; 3+ specific entities triggers `knowledge_dependent`).

E.3 Stage 3: Agent-Based Multi-View Reasoning

All specialized agents receive the query text T_q, contextual analysis O_{plan} from Stage 2, and can reference the similarity analysis S_{analysis} and rationale-augmented exemplars $\mathcal{C}_{\text{aug}}$ when needed.

E.3.1 Semantic Agent Prompt

This agent (A_{sem}) performs semantic analysis.

Semantic Agent

```
As a semantic analysis expert, deeply analyze the semantics of the text:
Text: "{text}" Context: "{context}" (if available)
Analysis: 1. Literal meaning and deep meaning 2. Emotional tendency and intensity (positive/negative/neutral, intensity 1-10) 3. Emotional coloring of key vocabulary 4. Overall tone (formal/informal/teasing/serious, etc.)
Return precise, concise structured analysis in several sentences.
```

E.3.2 Expectation Agent Prompt

This agent (A_{exp}) builds expectation models using contextual grounding from Meta-Planner.

Expectation Agent

```
Build expectation model using prior analysis:
Text: "{text}"
Analysis from prior agents: - Context analysis (from Context Agent): {context_agent_output} - Semantic analysis (from Semantic Agent): {semantic_agent_output} - Contextual grounding (from Meta-Planner): {meta_planner_context}
Build expectation model: 1. Normal expectation: What would be typical expression in this context 2. Actual vs. expected: Compare query expression with baseline expectation 3. Deviation analysis: Degree (1-10) and type (semantic/tonal/pragmatic) 4. Intentionality: Whether deviation appears deliberate for rhetorical effect
Reference Meta-Planner's expectation baseline for contextual grounding. Return structured analysis (3-4 sentences).
```

E.3.3 Knowledge Agent Prompt

This agent (A_{know}) retrieves background knowledge.

Knowledge Agent

```
As a knowledge retrieval expert, analyze entities and concepts in the text:
Text: "{text}" Context: {context_agent_output}
Please retrieve and analyze: 1. Key entities, people, events mentioned in the text 2. Common evaluations, stereotypes, public perceptions of these entities 3. Related background knowledge and common sense 4. Importance of this knowledge for understanding text sarcasm
Return precise, concise structured analysis in several sentences.
```

E.3.4 Alignment Agent Prompt

This agent (A_{align}) checks alignment between text and knowledge, referencing similarity patterns from Stage 1.

Alignment Agent

```
Check alignment between text and background knowledge:
Text: "{text}" Semantic analysis: {semantic_agent_output} Background knowledge: {knowledge_agent_output} Similarity patterns (from Stage 1): {similarity_analysis_summary}
Analyze: 1. Consistency: Does literal meaning align with background knowledge/common sense 2. Distortion detection: Deliberate misrepresentation or counter-factual claims 3. Inconsistency assessment: Degree (1-10) and nature (factual/evaluative/tonal) 4. Sarcastic intent: Whether inconsistency serves ironic/mocking purpose
Consider similarity patterns - if query matches known sarcastic patterns with knowledge contradictions, weight this evidence. Return structured analysis (3-4 sentences).
```

E.3.5 Incongruity Agent Prompt

This agent (A_{incon}) detects inconsistencies.

Incongruity Agent

```
Specifically detect and quantify inconsistencies:
Text: "{text}" Expectation analysis: {expectation_agent_output}
Please detect: 1. Specific types of inconsistency (semantic, emotional, logical, common sense, etc.) 2. Quantification of inconsistency degree (1-10) 3. Whether inconsistency has sarcastic effect 4. Key inconsistency points location and manifestation
Return precise, concise structured analysis in several sentences.
```

E.3.6 Rhetoric Agent Prompt

This agent (A_{rhet}) identifies rhetorical devices. Added constraints to reduce false positives from over-interpretation.

Rhetoric Agent

```
Identify if there is any rhetorical devices in the text:
Text: "{text}" Context: "{context}" (if available)
Focus on identifying: 1. Irony, exaggeration, understatement 2. Puns, rhetorical questions, interrogative sentences 3. Contrast, analogy 4. Contribution of these rhetorical devices to the expression 5. Do not think too much, these may be replies to other people on social media 6. Judge it as you are a normal linguist, not a sarcasm detector
Return precise, concise structured analysis in several sentences.
```

Modification Rationale: Points 5-6 instruct the agent to avoid over-analyzing casual social media language, reducing false positive rate (see Section 4.7.1 error analysis).

E.4 Stage 4: Synthesis and Final Judgment

E.4.1 Integrator Prompt

Enhanced with explicit decision logic, strict standards, and integration of Stage 1 similarity analysis to address over-prediction bias.

Integrator

```
As sarcasm expert, synthesize all evidence and make final judgment:
**Query:** "{text}"
**Similarity Analysis (Stage 1):** {similarity_analysis_from_stage1}
**Meta-Planner Output (Stage 2):** Selected plan: {plan_type}, Contextual analysis: {O_plan_summary}
**Agent Outputs (Stage 3):** {all_agent_outputs_json}
**Decision Framework:**
```

Positive Indicators: Contradiction literal/intended; rhetorical questions with mocking; exaggeration/understatement; emotional incongruity; ironic praise; expectation violations; strong similarity to sarcastic exemplars
Negative Indicators: Direct criticism without irony; genuine questions; consistent tone; literal matches intent; strong similarity to non-sarcastic exemplars
Synthesis Requirements: 1. Weight similarity analysis heavily - if Stage 1 shows strong pattern match to sarcastic/non-sarcastic, prioritize this evidence 2. Evaluate coherence: Do multiple agents agree on sarcastic interpretation? 3. Assess intentionality: Are contradictions/devices deliberate rhetorical strategy? 4. Context alignment: Does judgment match Meta-Planner's contextual inference?
STRICT Decision Logic: - Multiple clear indicators (irony+contradiction+mocking) + similarity to sarcastic → sarcastic - Strong similarity to sarcastic patterns + agent consensus on irony → sarcastic - Obvious contradiction literal/intended with intentionality → sarcastic - Direct criticism/opinions without ironic devices → non-sarcastic - Genuine inquiry (not mockery) + similarity to non-sarcastic → non-sarcastic - Emotional expression without contradiction + non-sarcastic patterns → non-sarcastic
Domain Context: Forum posts/comments/replies. Apply STRICT standards - only classify sarcastic with CLEAR evidence (intentional irony/mockery/contradiction). Casual criticism without ironic intent → non-sarcastic.
Output Format: Line 1: «LABEL» 1 (sarcastic) or 0 (non-sarcastic) Line 2: JSON {"label": 1/0, "conf": 0-1, "reasoning": "synthesis referencing similarity analysis, agent findings, and decision rationale"}