

Tool-meme: Tool-Learning Framework for Multimodal Meme Detection

Shuai Liu
Beijing Jiaotong University
Beijing, China
25140909@bjtu.edu.cn

Yingying Yao*
Beijing Jiaotong University
Beijing, China
yyyao@bjtu.edu.cn

Abstract

With the explosive growth of social media, multimodal memes (text and images) have become a key carrier of misinformation dissemination and emotional manipulation because they strongly depend on cultural background, satirical metaphor, and current events. Existing detection methods mostly rely on end-to-end deep learning models, but they face two fundamental limitations: first, static models are difficult to capture the dynamic semantic evolution of memes; second, fixed architectures cannot adapt to meme heterogeneity. This work proposes Tool-meme (**Tool-Learning Framework for Multimodal Meme Detection**), an LMM (Large Multimodal Model)-driven tools-learning framework. The framework uses multimodal representations only as input features, dynamically orchestrates tool chains through retrieval-augmented generation and meta-cognitive planning, and uses reinforcement learning to maximize tool-use efficiency. On HarM, FHM, and MAMI, Tool-meme outperforms all evaluated direct-LMM and agentic baselines under frozen-executor setting. Additional transfer and deployment analyses show consistent gains across tested executor families while lowering tool calls and cost vs. exhaustive execution, with explicit evidence chains for inspection. Code is available at <https://github.com/zizeLin/Tool-meme>.

1 Introduction

Memes have become one of the most active forms of multimodal communication on social media, and their harmful uses have motivated a line of benchmarks and detection studies for hate, misogyny, and broader social abuse (Kiela et al., 2020; Pramanick et al., 2021a; Lu et al., 2026; Fersini et al., 2022). A harmful meme is rarely harmful because of an isolated word or object; its meaning often emerges from the interaction between visual content, embedded text, cultural background, and an

*Corresponding author.

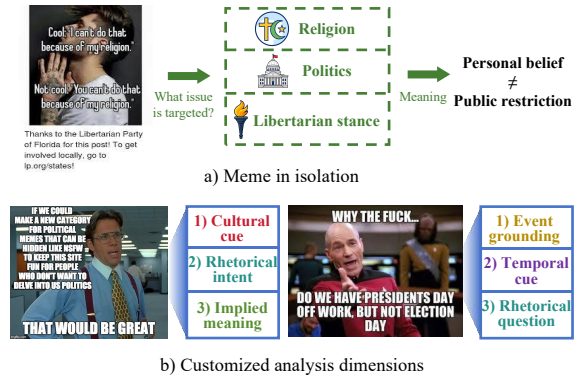


Figure 1: Illustration of motivation. The semantics of memes are highly dependent on implicit cultural context and are difficult to analyze in isolation. Different meme types require customized analysis dimensions.

implicit target (Sharma et al., 2022b,a; Lin et al., 2023). This makes meme detection different from ordinary image–text classification: a reused template, a recent public event, or an ironic caption can change the label without substantially changing the visible pixels (Lin et al., 2024; Mei et al., 2024, 2025). Consequently, effective harmful meme detection requires not only multimodal perception, but also context recovery and evidence-sensitive reasoning over why a particular image–text pair becomes harmful in a specific social situation (Lin et al., 2025, 2026; Liu et al., 2025).

Existing methods have made steady progress by improving cross-modal representation, extracting richer textual signals, and introducing external evidence into the detection process (Lu et al., 2019; Chen et al., 2020; Radford et al., 2021). Prompting, caption probing, and rationale distillation help expose implicit semantics that are not directly visible from the surface meme content (Cao et al., 2022, 2023; Lin et al., 2023). Retrieval-augmented and adaptation-based methods further use similar memes or contextual memory to improve robustness under domain shift and evolving

harmful expressions (Mei et al., 2024, 2025; Liu et al., 2025). Recent LMM-based systems introduce debate, question answering, and multi-agent collaboration to many tasks (Lu and Cui, 2026; Li and Lu, 2026), and making harmfulness judgments more interpretable and evidence-aware (Lin et al., 2024, 2025, 2026). However, these advances leave two fundamental limitations unresolved: first, static models are difficult to capture the dynamic semantic evolution of memes; second, fixed architectures cannot adapt to meme heterogeneity, because different memes require different combinations of grounding, cross-modal checking, rhetorical interpretation, and decision fusion (Sharma et al., 2022b; Liu et al., 2025; Mei et al., 2026).

This fixed-workflow assumption creates a mismatch with real meme streams. New memes appear quickly, but their decisive evidence is rarely uniform. Some cases hinge on image–text contradiction, while others require a recent event, a reused template, a target-group convention, or a subtle rhetorical inversion. As illustrated in Figure 1, different meme types need different analysis dimensions. An easy case should not pay the cost of every tool, and a hard case should not be forced to stop before the missing evidence is checked.

Motivated by this observation, we propose Tool-meme (**Tool**-Learning Framework for Multimodal **Meme** Detection). Tool-meme treats detection as adaptive evidence acquisition rather than one-pass classification. It first retrieves socio-cultural memory for the target meme, then uses a meta-cognitive planner to identify which semantic layers should be inspected. The resulting task graph is executed by an RL-guided adaptive tool router that selects LMM (Large Multimodal Model)-callable tools under dependency and budget constraints. Finally, Tool-meme fuses the acquired evidence into a calibrated decision, so that the model can spend computation on the semantic operations that the current meme actually requires.

Our contributions can be summarized as follows:

- We propose Tool-meme, a tool-learning framework that improves context-aware harmful meme detection through instance-specific evidence acquisition.
- We design two core modules, Meta-Cognitive Planning (MCP) and RL-Guided Adaptive Tool Routing (ATR), enabling the agent to diagnose meme-specific semantic layers, construct executable tool DAGs, and selectively

orchestrate LMM-callable tools under budget constraints.

- Extensive experiments on three public datasets show that Tool-meme outperforms the evaluated direct-LMM and agentic baselines under the frozen-executor setting. Transfer, executor-portability, and deployment analyses further characterize its accuracy–cost trade-off and evidence transparency.

2 Related Work

Multimodal Meme Detection Harmful meme detection has become an important research problem with the development of multimodal benchmarks for hateful, harmful, and misogynistic memes (Kiela et al., 2020; Pramanick et al., 2021a; Fersini et al., 2022). Early progress mainly came from pretrained vision-language encoders and multimodal Transformers, which improve image–text alignment and provide stronger representations for downstream classification (Lu et al., 2019; Chen et al., 2020; Radford et al., 2021). Subsequent work converts memes into richer textual or reasoning signals through prompting, caption probing, and rationale distillation, making implicit semantics more accessible to classifiers (Cao et al., 2022, 2023; Lin et al., 2023). Retrieval-guided methods introduce similar examples, external memory, or adaptation signals to handle the fact that harmful meanings change with events, templates, and social conventions (Mei et al., 2024, 2025; Liu et al., 2025). Recent LMM-based harmful meme detectors further use debate, targeted questioning, or explanation-then-detection to improve interpretability and robustness (Lin et al., 2024, 2025, 2026). These studies show that context matters, but many of them still rely on static representations or fixed inference procedures, which makes dynamic semantic evolution difficult to track once the meme stream changes.

Multi-Agent Systems Agent systems have shown that complex reasoning can be decomposed into interacting roles, executable tools, and iterative memory rather than being compressed into a single direct prediction (Wu et al., 2023; Hong et al., 2024; Chen et al., 2024b; Park et al., 2023; Li and Lu, 2026). Reasoning-oriented methods improve intermediate deliberation through chain-of-thought, self-consistency, tree-structured search, and consensus among multiple models (Wei et al., 2022;

Wang et al., 2023; Yao et al., 2023a; Chen et al., 2024a). Tool-augmented agents connect reasoning to external actions, while tool-learning or agent-optimization methods train models or policies to decide when tools, communication, or preference-optimized behavior should be used (Yao et al., 2023b; Schick et al., 2023; Wang et al., 2025; Chen et al., 2025). These training-oriented agent methods are relevant to Tool-meme because they move beyond prompting-only frameworks, but they often require large interaction traces, specialized optimization objectives, or expensive search and collaboration procedures (Schick et al., 2023; Li et al., 2026; Wang et al., 2025; Xu et al., 2025; Chen et al., 2025). In misinformation and fact-checking, agent systems have been used for logical decomposition, causal verification, multi-source tool use, and harmful meme reasoning (Ma et al., 2025; Cui et al., 2026; Lin et al., 2025; Liu et al., 2025). Overall, prior work motivates adaptive agents, but harmful meme detection still needs a lightweight mechanism that selects evidence operations according to meme-specific heterogeneity.

3 Method

Tool-meme formulates harmful meme detection as context recovery and evidence routing. Given a target meme, it first retrieves socio-cultural memory, then asks a meta-cognitive planner to decide which semantic layers need verification, routes only the necessary LMM-callable tools under a task DAG and budget, and finally fuses the resulting evidence into a calibrated decision. Figure 2 shows the complete pipeline.

3.1 Problem Formulation

Given a multimodal meme $x = (I, T)$, where I is the image and T is the overlaid text, OCR text, or user caption, the task is to predict harmfulness in the binary label space $\mathcal{Y} = \{0, 1\}$, where 1 denotes HARMFUL. Tool-meme can access an external multimodal meme memory $\mathcal{K} = \{c_i\}_{i=1}^N$. Each memory item may contain text, image description, OCR, label prior, event timestamp, cultural tags, and source metadata.

The system uses a tool library $\mathcal{T} = \{\tau_1, \dots, \tau_8\}$ as LMM-callable cognitive primitives and returns a label, confidence, evidence chain, and review flag:

$$F_{\Theta}(x, \mathcal{K}, \mathcal{T}) = (\hat{y}, c_{\text{final}}, e_{\text{chain}}, r_{\text{review}}). \quad (1)$$

The goal is not to call every tool. Tool-meme learns to spend computation where the meme requires it:

a political meme may need event grounding, while a visually simple joke may only need cross-modal alignment and rhetorical reading. The cost-aware routing objective, reward terms, and training details are provided in Appendix A.2.

3.2 Context-Augmented Multimodal Retrieval

Memes often compress social events, cultural templates, and implicit targets into a short image-text pair. CAMR therefore provides the planner with a socio-cultural anchor before any tool is selected. Frozen multimodal encoders (Radford et al., 2021; Li et al., 2022, 2023) encode the target meme and retrieve similar memory items:

$$\begin{aligned} \mathbf{q}_x &= \text{norm}(W_q[f_v(I); f_t(T); f_v(I) \odot f_t(T)]), \\ \mathcal{C}_K &= \text{TopK}_{c_i \in \mathcal{K}} \mathbf{q}_x^\top \mathbf{e}_i. \end{aligned} \quad (2)$$

Here, $\mathbf{e}_i$ is the indexed key of memory item c_i , and $\mathcal{C}_K$ contains the most similar contexts. CAMR then packs the retrieved items into a compact context package:

$$\begin{aligned} \rho_K, \kappa_K, m_K, \mathcal{P}_K &= \Phi_{\text{ctx}}(\mathcal{C}_K, \mathcal{S}_K), \\ \mathcal{S}_K &= \{s_{(k)}, y_{(k)}, \mu_{(k)}\}_{k=1}^K. \end{aligned} \quad (3)$$

ρ_K is the retrieved label prior, κ_K is retrieval reliability, and $m_K \in \{\text{context}, \text{no-context}\}$ indicates whether the retrieval should be treated as strong context or weak background. The package $\mathcal{P}_K$ includes retrieved text, image descriptions, labels, event contexts, and cultural tags. When $m_K = \text{no-context}$, downstream modules can read the retrieved content but cannot use neighbor labels as strong evidence. Memory-bank construction is detailed in Appendix Section D.

3.3 Meta-Cognitive Planning

MCP is the decision center that prevents Tool-meme from using a fixed tool chain for every meme. Let Ω denote the predefined semantic-layer inventory available to MCP. It reads the target meme and $\mathcal{P}_K$, diagnoses which semantic layers are relevant, and converts them into an executable task graph:

$$\begin{aligned} \Omega_x &= M_{\theta}(x, \mathcal{P}_K) \subseteq \Omega, \\ G_x &= \mathcal{R}_{\text{fix}}(\Pi_{\theta}(x, \mathcal{P}_K, \Omega_x)) = (\mathcal{V}_x, \mathcal{E}_x, \tau, \omega), \\ \mathcal{F}_0 &= \{v \in \mathcal{V}_x \mid \text{Pred}(v) = \emptyset\}. \end{aligned} \quad (4)$$

Ω_x is the subset of Ω selected by MCP for sample x . This inventory covers semantic layers such as event background, cross-modal conflict, irony, cultural

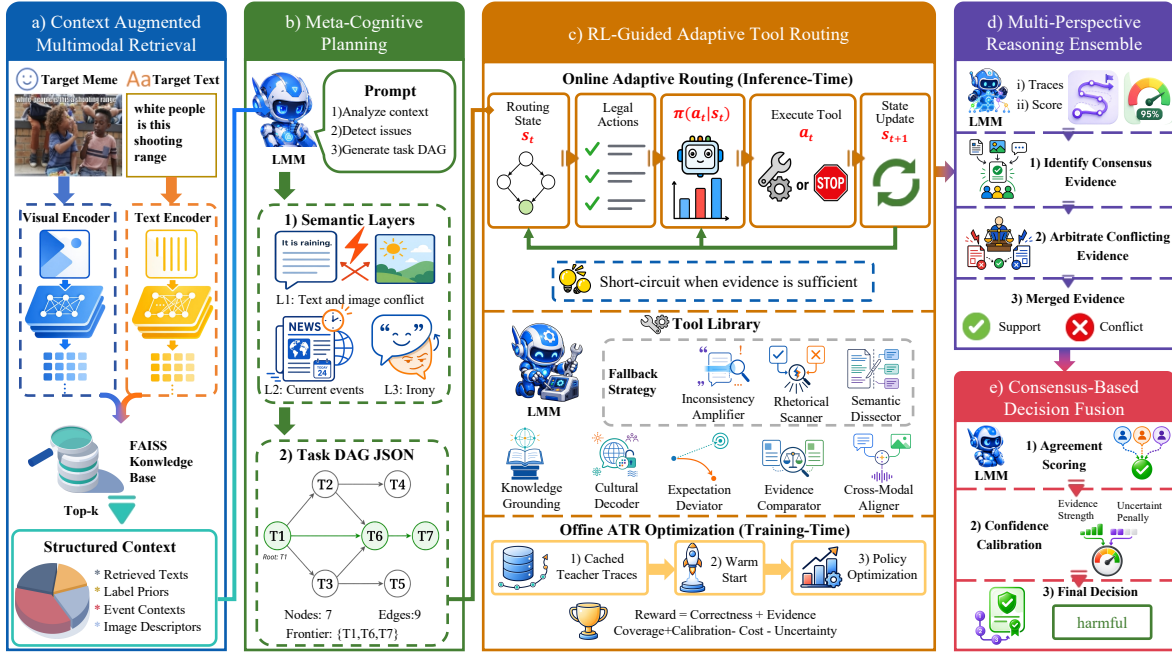


Figure 2: Overview of Tool-meme. The framework contains five coordinated panels: (a) Context-Augmented Multimodal Retrieval retrieves socio-cultural memory for a target meme; (b) Meta-Cognitive Planning generates an instance-specific task DAG; (c) RL-Guided Adaptive Tool Routing executes feasible tools under DAG and budget constraints; (d) Multi-Perspective Reasoning Ensemble synthesizes tool traces; and (e) Consensus-Based Decision Fusion produces a calibrated prediction with an interpretable evidence chain.

template, and target group. A node v represents an analysis operation assigned to tool $\tau(v)$, an edge represents a semantic dependency, and $\mathcal{F}_0$ is the set of executable starting nodes. For example, a meme tied to a public event activates knowledge grounding before inconsistency checking, whereas a context-poor meme prioritizes visible image-text evidence. Schema repair only makes the decoded plan executable; it does not add new semantic claims. The complete inference procedure is summarized in Algorithm 1, and the planning contract is given in Appendix Section I.

3.4 RL-Guided Adaptive Tool Routing

ATR is the core execution component of Tool-meme. It turns the MCP graph into an evidence-producing trajectory while avoiding unnecessary tool calls. The tool library contains eight cognitive primitives: semantic dissecting, rhetorical scanning, knowledge grounding, expectation deviation, cross-modal alignment, inconsistency amplification, cultural decoding, and evidence comparison. All tools share the same input-output schema, so the router can compose them without tool-specific control logic.

ATR operates as a DAG-constrained, risk-aware, and cost-sensitive router. MCP decides which se-

mantic layers should be checked and repairs them into G_x ; ATR learns how much additional evidence is worth acquiring on this graph. At step t , the lightweight policy π_ϕ observes the frontier, completed nodes, collected outputs, remaining budget, uncertainty, and tool-level conflict, and it selects from the executable frontier or the stop action:

$$s_t = (x, \mathcal{P}_K, G_x, \mathcal{F}_t, \mathcal{H}_t, \mathcal{O}_t, b_t, u_t, \chi_t),$$

$$a_t = \arg \max_{a \in \mathcal{F}_t \cup \{\text{STOP}\}} \pi_\phi(a | s_t). \quad (5)$$

The action mask keeps routing inside the MCP frontier and limits execution to predefined tools with satisfied dependencies. The STOP action is also gated by evidence risk; it is legal only when the current confidence is high, all mandatory layers are resolved, conflict is below threshold, and the decision has support beyond weak retrieval. If the policy selects a node v_t , ATR calls its assigned tool and updates the graph state:

$$o_t = \tau(v_t)(x, \mathcal{P}_K, \mathcal{O}_t) = (r_t, p_t, q_t, c_t),$$

$$\mathcal{O}_{t+1} = \mathcal{O}_t \cup \{o_t\},$$

$$\mathcal{H}_{t+1} = \mathcal{H}_t \cup \{v_t\}, \quad b_{t+1} = b_t - c_t,$$

$$\mathcal{F}_{t+1} = \{v \in \mathcal{V}_x \setminus \mathcal{H}_{t+1} \mid \text{Pred}(v) \subseteq \mathcal{H}_{t+1}\}. \quad (6)$$

Here, r_t is the reasoning trace, p_t is the local label signal or unknown, q_t is confidence, and c_t

is cost. When the acquired evidence is confident, consistent, and sufficient for the mandatory layers, ATR can short-circuit low-priority successors whose expected marginal gain is small. The resulting evidence set is $\mathcal{O}_x = \{o_t\}_{t=1}^L$, and the trajectory Γ_x records the selected actions. We train only the routing policy while keeping the GPT-4o, Claude, or Gemini tool executors frozen, using supervised warm-start followed by group-relative policy optimization in a cached tool environment. The reward favors correct, calibrated, evidence-using, layer-covering, and low-cost routes, with an explicit penalty for premature stopping. The executable inference and training details are given in Appendix Sections A.1 and A.2; the shared tool prompts are listed in Appendix Section I.

3.5 Multi-Perspective Reasoning Ensemble

MPRE receives $\mathcal{O}_x$ from ATR and converts local tool traces into a single evidence state. It weighs each trace by confidence, semantic relevance, and cost, then asks an LMM synthesizer to separate supporting, opposing, and unresolved evidence:

$$\begin{aligned}\alpha_t &= \text{softmax}_t(\eta_q q_t + \mu_d d_t - \nu_c c_t), \\ \chi_x &= \text{Dis}(\{(\alpha_t, p_t)\}_{t=1}^L), \\ S_{\text{con}} &= E_\omega(\mathcal{O}_x, \alpha, \chi_x, \mathcal{P}_K).\end{aligned}\quad (7)$$

Here, d_t measures relevance between a tool output and the semantic layer it verifies, with $\eta_q = 1.0$, $\mu_d = 1.0$, and $\nu_c = 0.5$ by default. χ_x summarizes disagreement among weighted local judgments, and S_{con} is the consensus summary. This summary explicitly records harmful evidence, harmless evidence, and unresolved uncertainty. The structured synthesis contract is given in Appendix Section I.

3.6 Consensus-Based Decision Fusion

CBDF combines the MPRE consensus with retrieval reliability and routing diagnostics. As the last decision-fusion module in the system mapping F_Θ of Eq. (1), J_ω receives only structured evidence, not free-form intermediate thoughts, and produces the final inspectable output:

$$\begin{aligned}\mathbf{z}_x^{\text{dec}} &= (S_{\text{con}}, \kappa_K \rho_K, m_K, \chi_x, \bar{q}_x, \Delta_q), \\ \hat{\mathbf{d}}_x &= J_\omega(\mathbf{z}_x^{\text{dec}}) = (\hat{y}, c_{\text{final}}, e_{\text{chain}}, r_{\text{review}}).\end{aligned}\quad (8)$$

$\hat{\mathbf{d}}_x$ is the sample-level instantiation of the output tuple defined in Eq. (1). $\bar{q}_x$ is the weighted average tool confidence, and Δ_q measures confidence dispersion. High conflict, large confidence split, or

weak-retrieval low-confidence judgment can trigger the review flag, but these rules do not overwrite the predicted label. The final JSON output contains the label, confidence, concise reasoning explanation, evidence chain, and review flag. The decision-output contract is given in Appendix Section I.

4 Experiments

In this section, we conduct extensive experiments to verify the effectiveness of Tool-meme. We first outline the datasets, baselines, and evaluation metrics. Detailed settings, cross-dataset transfer, adaptive routing, and human evaluation are provided in Appendix Sections B–F; additional transfer, controlled ablations, memory-bank auditing, and deployment measurements are reported in Appendix Sections G–H.

4.1 Experimental Setup

4.1.1 Datasets

To evaluate the effectiveness of the proposed Tool-meme, we conduct comprehensive experiments on three public meme benchmarks, including HarM (Pramanick et al., 2021a; Sharma et al., 2022b; Liu et al., 2025), FHM (Kiela et al., 2020), and MAMI (Fersini et al., 2022). Detailed dataset statistics are provided in Appendix B. Following existing work (Liu et al., 2025), we merge partially harmful and very harmful samples into the positive class.

4.1.2 Baselines

We compare Tool-meme with 9 competitive baselines, which can be divided into three categories: (1) **Closed-Source LMM**, including GPT-4o¹, Gemini-1.5-Flash², and Claude-3.5-Sonnet³; (2) **Open-Source LMM**, including LLaVA-1.6-34B, Qwen2.5-VL-72B, and Qwen3-VL-32B-Instruct (Liu et al., 2024; Bai et al., 2025b,a); and (3) **Agent-Based Methods (Closed-Source)**, including MIND (GPT-4o), A-A-U (GPT-4o), and ExplainHM++ (GPT-4o) (Liu et al., 2025; Lin et al., 2025, 2026). All baselines use their method-specific inference protocols under the same binary label space.

4.1.3 Evaluation Metrics

Following prior work (Sharma et al., 2022b; Liu et al., 2025), we use the following metrics to evalu-

¹<https://openai.com/index/hello-gpt-4o/>

²<https://blog.google/technology/ai/google-gemini-update-flash-ai-assistant-io-2024/>

³<https://www.anthropic.com/news/claude-3-5-sonnet>

ate Tool-meme: classification accuracy (ACC) and macro F1 score (M-F1).

4.2 Overall Performance (RQ1)

To evaluate Tool-meme under the frozen-executor setting, we compare it with 9 competitive baselines on the three datasets and report the results in Table 1. From these results, we make the following observations:

(1) Direct LMM prediction provides a usable foundation. Direct multimodal large models use general multimodal knowledge acquired during pretraining. Closed-source models and open-source LMMs both achieve usable results on some datasets, indicating that general LMMs already have some ability to recognize harmful semantics. However, Tool-meme outperforms these direct-prediction baselines because it does not rely only on surface image-text cues in the input.

(2) Agent-based closed-source methods are stronger than most direct prediction baselines. MIND, A-A-U, and ExplainHM++ improve the detection results by inserting retrieval, question answering, or debate-style reasoning between the meme and the final decision. ExplainHM++ is the strongest baseline in Table 1, showing the benefit of closed-source agentic reasoning for harmful meme detection. However, Tool-meme outperforms these frameworks because it adds adaptive learnable tool routing rather than executing a fixed inference protocol.

(3) Tool-meme surpasses the evaluated agentic baselines on all three datasets. Compared with the strongest matched agentic baseline ExplainHM++, Tool-meme improves ACC by 2.80, 2.22, and 2.80 percentage points on HarM, FHM, and MAMI, respectively, and improves Macro-F1 by 2.87, 2.26, and 2.88 percentage points, respectively. The gains are positive for every dataset and every reported metric, giving average ACC and Macro-F1 improvements of 2.61 and 2.67 percentage points. This result shows a consistent gain within the evaluated frozen-executor/agentic setting rather than an absolute state-of-the-art claim. Stronger task-specific fine-tuned and prompt-tuned HarM systems are reported separately as non-matched references in Appendix Section G.1.

4.3 Ablation Studies (RQ2)

To evaluate the contribution of each core component to Tool-meme performance, we design the following five model variants: (1) w/o CAMR: re-

moving context-augmented multimodal retrieval; (2) w/o MCP: removing the meta-cognitive planner; (3) w/o RL-ATR: removing the adaptive tool routing module; (4) w/o MPRE: removing the multi-perspective reasoning ensemble module; and (5) w/o CBDF: removing the consensus-based decision fusion module. Table 2 lists the experimental results, revealing the following three observations:

(1) Removing any core component causes performance degradation. Specifically, CAMR provides socio-cultural context missing from the target image and text; MCP transforms retrieval and input cues into a sample-specific semantic-layer plan; RL-ATR selects tools to execute according to evidence coverage and budget; MPRE synthesizes heterogeneous tool traces into an inspectable evidence state; and CBDF performs calibrated judgment under conflict and uncertainty. Therefore, removing any module breaks the complete reasoning chain from retrieval, planning, routing, and evidence synthesis to final decision.

(2) Removing CAMR and RL-ATR leads to the largest Macro-F1 drops. Averaged over the three datasets, removing RL-ATR lowers Macro-F1 by 3.80 points and removing CAMR lowers it by 3.16 points. This observation supports that external socio-cultural memory is a key source for handling context sparsity in harmful memes, and that tool execution must be adaptively routed according to evidence coverage and instance uncertainty.

(3) Evidence synthesis and decision fusion jointly ensure the effectiveness of tool reasoning. Both w/o MPRE and w/o CBDF cause consistent declines, indicating that MPRE and CBDF mitigate the influence of noise from individual tools or local evidence conflicts.

4.4 Parameter Sensitivity (RQ3)

To evaluate the robustness of the framework to practical configuration choices, we vary the retrieval size $K \in \{1, 3, 5, 10, 15, 20\}$, the short-circuit threshold $\tau_{sc} \in \{0.60, 0.70, 0.80, 0.90, 0.95\}$, and the number of available tools. Figure 4 reports the corresponding sensitivity curves. Each experiment changes only one factor while keeping the other settings at the default configuration. Overall, Tool-meme remains stable within practically usable configuration ranges. Too small a K leads to insufficient cultural and event context, while excessive retrieval introduces noisy neighbors. A moderate short-circuit threshold preserves evidence coverage and reduces redundant calls, whereas overly aggres-

Model	HarM		FHM		MAMI		Avg. F1
	Acc. ↑	M-F1 ↑	Acc. ↑	M-F1 ↑	Acc. ↑	M-F1 ↑	
<i>Closed-Source LMM</i>							
GPT-4o	67.51	60.29	68.80	68.25	81.00	80.82	69.79
Gemini-1.5-Flash	60.20	58.90	66.10	64.18	76.40	73.24	65.44
Claude-3.5-Sonnet	64.73	57.92	65.53	63.55	78.58	77.81	66.43
<i>Open-Source LMM</i>							
LLaVA-1.6-34B (Liu et al., 2024)	67.51	61.59	64.00	63.51	71.30	71.28	65.46
Qwen2.5-VL-72B (Bai et al., 2025b)	69.37	63.18	71.27	71.02	78.66	77.52	70.57
Qwen3-VL-32B-Instruct (Bai et al., 2025a)	66.20	65.67	67.65	67.32	74.76	72.25	68.41
<i>Agent-Based Methods (Closed-Source)</i>							
MIND (GPT-4o) (Liu et al., 2025)	68.73	65.92	70.53	68.55	83.54	83.01	72.49
A-A-U (GPT-4o) (Lin et al., 2025)	67.61	66.21	70.14	69.57	81.70	81.32	72.37
ExplainHM++ (GPT-4o) (Lin et al., 2026)	<u>70.84</u>	<u>69.50</u>	<u>75.81</u>	<u>74.72</u>	<u>84.01</u>	<u>83.86</u>	<u>76.03</u>
Tool-meme(GPT-4o)	73.64	72.37	78.03	76.98	86.81	86.74	78.70

Table 1: Main results on HarM, FHM, and MAMI. Harmful meme detection results on three datasets. The accuracy and macro-averaged F1 scores (%) are reported as the metrics. Baselines are grouped by model access and inference protocol under the same binary label space. Best and second-best results are shown in **bold** and underlined.

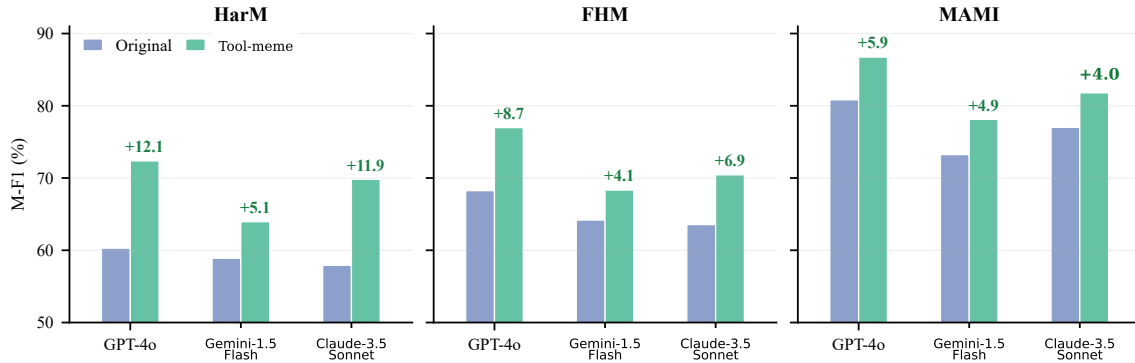


Figure 3: M-F1 improvements across LMM backbones. The figure reports M-F1 on HarM, FHM, and MAMI before and after applying Tool-meme to each backbone. Green annotations denote absolute M-F1 gains.

Var.	HarM		FHM		MAMI	
	Acc. ↑	M-F1 ↑	Acc. ↑	M-F1 ↑	Acc. ↑	M-F1 ↑
Tool-meme	73.64	72.37	78.03	76.98	86.81	86.74
w/o CAMR	71.26	69.82	74.08	73.90	83.09	82.90
w/o MCP	72.23	71.91	76.97	73.87	84.80	84.65
w/o RL-ATR	72.89	70.54	74.79	71.65	83.63	82.49
w/o MPRE	72.69	71.28	76.33	74.22	84.09	83.93
w/o CBDF	72.82	72.16	77.52	74.40	85.25	85.11

Table 2: Module ablation on HarM, FHM, and MAMI. Scores are Accuracy and M-F1 (%).

sive stopping removes still useful reasoning steps. Increasing the tool library brings gains before the major semantic layers are covered, after which the marginal gain becomes small.

4.5 Compatibility Study (RQ4)

To study executor portability with respect to LMM scale and type, we select three representative proprietary multimodal backbones: GPT-4o, Gemini-1.5-Flash, and Claude-3.5-Sonnet. The experimental results are shown in Figure 3. Tool-meme gives positive gains for all three evaluated backbones, while GPT-4o maintains the highest absolute performance after integration. These results show that the framework is not tied to a single evaluated proprietary executor; they do not imply universal model-agnostic behavior. Appendix Section G.6 further reports matched end-to-end experiments with three open-source executors.

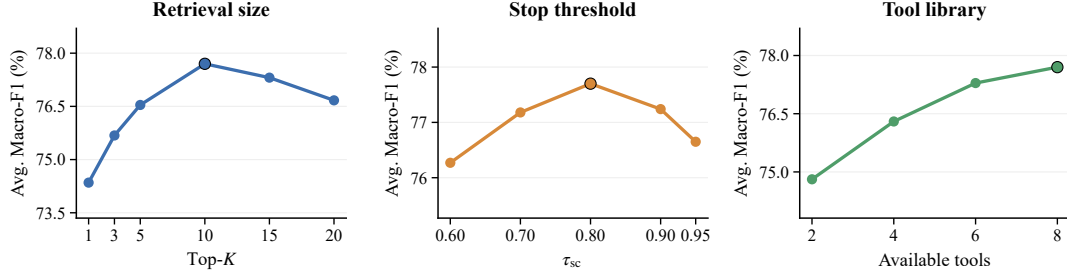


Figure 4: Parameter sensitivity analysis. Sensitivity to retrieval size, short-circuit threshold, and tool-library size.

	Case A	Case B
Meme		
Text	“bravery at its finest”	“white privilege discount”
Actual	harmful	harmful
LMM	harmless ×	harmless ×
Tool-meme	harmful ✓	harmless ×
Outcome	Correct	Incorrect
Cost	Calls: 6/6 Shortcut: 0	Calls: 1/4 Shortcut: 3

Table 3: Case study of Tool-meme interpretability and efficiency. The table compares a successful full-route case and a failure caused by early short-circuiting.

4.6 Case Study and Explainability (RQ5)

To analyze the explainability of Tool-meme, we select one successful case and one failed case for comparative discussion. This case study focuses on how Tool-meme decides whether to continue executing tools according to sample-level uncertainty, and how this adaptive routing affects the final prediction. Table 3 presents the two cases in the same half-column table, marking correct prediction and failure boundary with a green check mark and a red cross.

Case A demonstrates the advantage of Tool-meme on high-conflict samples. The text superficially praises “bravery”, but the image context forms an ironic image-text conflict; the direct LMM outputs harmless, while Tool-meme preserves the

complete tool path and rechecks irony, inconsistency, and cross-modal evidence, finally correctly identifying the sample as harmful. Case B reveals the boundary of the efficiency mechanism: the early textual judgment around “white privilege discount” triggers short-circuiting, and ATR compresses a four-step path into a single call, saving 75% of tool calls; however, the skipped later checks could have supplemented the context of group harm, causing Tool-meme to still output harmless. This comparison shows that the explanation of Tool-meme comes not only from the final label, but also from the actual tool cost and the evidence-coverage difference caused by short-circuiting.

5 Conclusion

We presented Tool-meme, a tool-learning framework for multimodal harmful meme detection. Tool-meme retrieves socio-cultural memory, plans meme-specific semantic checks, and routes only the necessary LMM-callable tools before fusing the resulting evidence into a calibrated decision. Under the frozen-executor setting, experiments on HarM, FHM, and MAMI show consistent improvements over the evaluated direct-model and agent-based baselines, while controlled routing and deployment measurements demonstrate a favorable accuracy–cost trade-off. Transfer diagnostics and explicit evidence chains further clarify where the framework helps and where its claims remain bounded. These results suggest that adaptive evidence acquisition is a practical direction for context-aware harmful-content detection without requiring task-specific fine-tuning of the LMM executor.

Ethical Considerations

Tool-meme raises ethical considerations because it studies harmful memes and relies on LMM-based reasoning over sensitive multimodal content.

Data, rights, and privacy: We use public research benchmarks and construct the memory bank only from registered sources that pass license/terms, research-indexing, privacy, and access-control checks. The registry records the collection window, terms snapshot, redistribution permissions, and a contact or takedown channel. Sources with unclear permission or private content are excluded. Usernames, avatars, URLs, geolocation, and private identifiers are removed or hashed, and original media are not redistributed when the applicable permission allows only derived features or metadata. Test samples, their labels, and exact or near-duplicate clusters are blocked from retrieval construction.

Limitations

Despite the improvements presented by Tool-meme, our study has four main limitations:

Memory coverage: Tool-meme depends on the coverage, timeliness, and cultural breadth of the retrieval memory. Memes grounded in local events, emerging templates, community-internal irony, or platform-specific slang may remain under-contextualized when relevant evidence is missing. Retrieved neighbors may also introduce biased label priors or misleading cultural associations, even though Tool-meme calibrates retrieval reliability.

Routing trade-off: The adaptive router reduces redundant tool calls by stopping when evidence appears stable. However, early short-circuiting may skip later checks that would reveal group-targeted harm, cross-modal conflict, or subtle coded meanings. This risk is most salient for hateful, misogynistic, or political memes whose decisive evidence only emerges after multiple semantic layers are inspected.

Evaluation and training-regime scope: The main experiments follow a unified binary harmfulness setting on HarM, FHM, and MAMI. Appendix G adds Multi³Hate and MAMI Subtask B as frozen-router transfer diagnostics, but these results remain limited to the evaluated languages, locales, and four MAMI harm types; they do not cover all target groups, speaker intents, or moderation policies. Task-specific fine-tuned HarM systems also achieve higher absolute accuracy than Tool-meme. Our claim is therefore restricted to the evaluated frozen-executor setting and its accuracy–cost/evidence trade-off. Tool-meme still depends on OCR, visual grounding, LMM calibration, and

prompt-specialized tool outputs, which may propagate errors into the final evidence chain.

Emerging-meme coverage: The registered-tool interface could accommodate a streaming retrieval or novelty-detection tool, but no such tool is implemented or evaluated here. Timestamped memory entries, temporal retrieval restrictions, no-context mode under weak retrieval, and review flags mitigate unsupported use of stale context; they do not establish real-time detection of genuinely new templates or events. Evaluating that capability requires a temporally split benchmark and remains future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Youth Program) under Grant No. 62302033.

References

- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhi-fang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. [Qwen3-vl technical report](#). Preprint, arXiv:2511.21631.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. [Qwen2.5-vl technical report](#). arXiv preprint arXiv:2502.13923.
- Minh Duc Bui, Katharina von der Wense, and Anne Lauscher. 2025. [Multi³Hate: Multimodal, multilingual, and multicultural hate speech detection with vision–language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9714–9731, Albuquerque, New Mexico. Association for Computational Linguistics.
- Rui Cao, Ming Shan Hee, Adriel Kuek, Wen-Haw Chong, Roy Ka-Wei Lee, and Jing Jiang. 2023. [Procap: Leveraging a frozen vision-language model for hateful meme detection](#). In *Proceedings of the 31st ACM International Conference on Multimedia*, MM ’23, page 5244–5252, New York, NY, USA. Association for Computing Machinery.
- Rui Cao, Roy Ka-Wei Lee, Wen-Haw Chong, and Jing Jiang. 2022. [Prompting for multimodal hateful meme classification](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language*

- Processing*, pages 321–332, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024a. [ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.
- Sirui Chen, Mengshi Zhao, Lei Xu, Yuying Zhao, Beier Zhu, Hanwang Zhang, Shengjie Zhao, and Chaochao Lu. 2025. [Depo: Dual-efficiency preference optimization for llm agents](#). *Preprint*, arXiv:2511.15392.
- Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2024b. [Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors](#). In *The Twelfth International Conference on Learning Representations*.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. [UNITER: UNiversal image-TExt representation learning](#). In *Computer Vision – ECCV 2020*, pages 104–120. Springer International Publishing.
- Xing Cui, Yueying Zou, Zekun Li, Peipei Li, Xinyuan Xu, Xuannan Liu, and Huaibo Huang. 2026. [T2agent: A tool-augmented multimodal misinformation detection agent with monte carlo tree search](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 40:175–183.
- Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. 2022. [Semeval-2022 task 5: Multimedia automatic misogyny identification](#). In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 533–549. Association for Computational Linguistics.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xianwu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. [MetaGPT: Meta programming for a multi-agent collaborative framework](#). In *The Twelfth International Conference on Learning Representations*.
- Jeff Johnson, Matthijs Douze, and Herve Jegou. 2021. [Billion-Scale Similarity Search with GPUs](#). *IEEE Transactions on Big Data*, 7(03):535–547.
- Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. [The hateful memes challenge: Detecting hate speech in multimodal memes](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Chenhui Li and Weihai Lu. 2026. Decoding the market’s pulse: Context-enriched agentic retrieval augmented generation for predicting post-earnings price shocks. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3055–3073.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. [BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. [BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.
- Zhuofeng Li, Haoxiang Zhang, Seungju Han, Sheng Liu, Jianwen Xie, Yu Zhang, Yejin Choi, James Zou, and Pan Lu. 2026. [In-the-flow agentic system optimization for effective planning and tool use](#). In *The Fourteenth International Conference on Learning Representations*.
- Hongzhan Lin, Wei Gao, Jing Ma, Yang Deng, Ziyang Luo, Bo Wang, Ruichao Yang, and Tat-Seng Chua. 2026. [ExplainHM++: Explainable Harmful Meme Detection With Retrieval-Augmented Debate Between Large Multimodal Models](#). *IEEE Transactions on Knowledge & Data Engineering*, 38(02):1040–1053.
- Hongzhan Lin, Ziyang Luo, Wei Gao, Jing Ma, Bo Wang, and Ruichao Yang. 2024. [Towards explainable harmful meme detection through multimodal debate between large language models](#). In *Proceedings of the ACM Web Conference 2024, WWW ’24*, page 2359–2370, New York, NY, USA. Association for Computing Machinery.
- Hongzhan Lin, Ziyang Luo, Jing Ma, and Long Chen. 2023. [Beneath the surface: Unveiling harmful memes with multimodal reasoning distilled from large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9114–9128, Singapore. Association for Computational Linguistics.
- Xuanrui Lin, Chao Jia, Junhui Ji, Hui Han, and Usman Naseem. 2025. [Ask, acquire, understand: A multimodal agent-based framework for social abuse detection in memes](#). In *Proceedings of the ACM on Web Conference 2025, WWW ’25*, page 4734–4744,

- New York, NY, USA. Association for Computing Machinery.
- Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. [Llava-next: Improved reasoning, ocr, and world knowledge](#). Accessed: 2026-05-25.
- Ziyan Liu, Chunxiao Fan, Haoran Lou, Yuexin Wu, and Kaiwei Deng. 2025. [MIND: A multi-agent framework for zero-shot harmful meme detection](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 923–947, Vienna, Austria. Association for Computational Linguistics.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. [Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23.
- Wayne Lu and Xiaoxi Cui. 2026. Dealt: Llm-driven diversity-enhanced data augmentation for long-tail text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence, Volume 40, Number 38, volume 40*, pages 32338–32346.
- Weihai Lu, Zhejun Zhao, Yanshu Li, and Huan He. 2026. Mm-stancedet: Retrieval-augmented multimodal multi-agent stance detection. *arXiv preprint arXiv:2604.27934*.
- Jiatong Ma, Linmei Hu, Rang Li, and Wenbo Fu. 2025. [Local: Logical and causal fact-checking with llm-based multi-agents](#). In *Proceedings of the ACM on Web Conference 2025, WWW '25*, page 1614–1625, New York, NY, USA. Association for Computing Machinery.
- Jingbiao Mei, Jinghong Chen, Weizhe Lin, Bill Byrne, and Marcus Tomalin. 2024. [Improving hateful meme detection through retrieval-guided contrastive learning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5333–5347, Bangkok, Thailand. Association for Computational Linguistics.
- Jingbiao Mei, Jinghong Chen, Guangyu Yang, Weizhe Lin, and Bill Byrne. 2025. [Robust adaptation of large multimodal models for retrieval augmented hateful meme detection](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23806–23828, Suzhou, China. Association for Computational Linguistics.
- Jingbiao Mei, Mingsheng Sun, Jinghong Chen, Pengda Qin, Yuhong Li, Da Chen, and Bill Byrne. 2026. [ExPO-HM: Learning to explain-then-detect for hateful meme detection](#). In *The Fourteenth International Conference on Learning Representations*.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. [Generative agents: Interactive simulacra of human behavior](#). In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST '23*, New York, NY, USA. Association for Computing Machinery.
- Shraman Pramanick, Dimitar Dimitrov, Rituparna Mukherjee, Shivam Sharma, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021a. [Detecting harmful memes and their targets](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2783–2796, Online. Association for Computational Linguistics.
- Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021b. [MOMENTA: A multimodal framework for detecting harmful memes and their targets](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4439–4455, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 8748–8763. PMLR.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. [Toolformer: Language models can teach themselves to use tools](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 68539–68551. Curran Associates, Inc.
- Shivam Sharma, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2022a. [DISARM: detecting the victims targeted by harmful memes](#). In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, Findings of ACL, pages 1572–1588. Association for Computational Linguistics.
- Shivam Sharma, Firoj Alam, Md. Shad Akhtar, Dimitar Dimitrov, Giovanni Da San Martino, Hamed Firooz, Alon Halevy, Fabrizio Silvestri, Preslav Nakov, and Tanmoy Chakraborty. 2022b. [Detecting and understanding harmful memes: A survey](#). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022*, pages 5597–5606, Vienna, Austria. International Joint Conferences on Artificial Intelligence Organization. Survey Track.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language](#)

models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. 2025. [Agent-Dropout: Dynamic agent elimination for token-efficient and high-performance LLM-based multi-agent collaboration](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24013–24035, Vienna, Austria. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2023. [Autogen: Enabling next-gen llm applications via multi-agent conversation](#). *Preprint*, arXiv:2308.08155.

Wenyan Xu, Dawei Xiang, Tianqi Ding, and Weihai Lu. 2025. Mmm-fact: A multimodal, multi-domain fact-checking dataset with multi-level retrieval difficulty. *arXiv preprint arXiv:2510.25120*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. [React: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations*.

A Method Supplementary Details

This appendix focuses on the two procedures needed to reproduce Tool-meme: end-to-end inference and RL-guided ATR training. The shared tool interface and role-specific prompts are provided in Appendix Section I.

A.1 End-to-End Tool-meme Inference

Algorithm 1 summarizes the complete inference path from context retrieval and task-graph construction to adaptive tool execution and calibrated decision fusion.

Algorithm 1 End-to-End Tool-meme Inference

Require: Meme $x = (I, T)$, memory bank $\mathcal{K}$, tool prompt set $\mathcal{T}$, budget B , router π_ϕ

Ensure: Prediction $\hat{y}$, confidence c , evidence chain E

- 1: Retrieve $\mathcal{C}_K$ with CAMR and compute $\mathcal{P}_K = (\rho_K, \kappa_K, m_K, \text{context})$
- 2: Ask MCP to decode semantic layers and a candidate DAG G'_x
- 3: Repair G'_x into executable DAG G_x ; use the minimal fallback path if repair fails
- 4: Obtain a no-tool fast judgment $(\hat{y}^{\text{fast}}, q^{\text{fast}})$
- 5: **if** $q^{\text{fast}} > \tau_{\text{sc}}$ and the STOP gate is satisfied **then**
- 6: **return** $(\hat{y}^{\text{fast}}, q^{\text{fast}}, \emptyset)$
- 7: **end if**
- 8: Initialize selected nodes $\mathcal{H} \leftarrow \emptyset$, outputs $\mathcal{O} \leftarrow \emptyset$, remaining budget $b \leftarrow B$
- 9: **while** $b > 0$ and executable unresolved nodes remain **do**
- 10: Build frontier $\mathcal{F} = \{v \notin \mathcal{H} \mid \text{Pred}(v) \subseteq \mathcal{H}\}$
- 11: Add STOP to the legal set only if the risk-aware gate is satisfied
- 12: Select $a \in \mathcal{F} \cup \{\text{STOP}\}$ using π_ϕ
- 13: **if** $a = \text{STOP}$ or $\mathcal{F} = \emptyset$ **then**
- 14: break
- 15: **end if**
- 16: Execute $\tau(a)$, validate its schema, and append valid or failed evidence to $\mathcal{O}$
- 17: Update $\mathcal{H}$, b , u , χ , and the frontier; short-circuit low-priority successors if evidence is sufficient
- 18: **end while**
- 19: Use MPRE to synthesize $\mathcal{O}$ into evidence state E
- 20: Use CBDF to combine E , $\mathcal{P}_K$, and routing diagnostics into $(\hat{y}, c)$
- 21: **return** $(\hat{y}, c, E)$

A.2 RL-Guided ATR Training

ATR training adapts group-relative tool-use learning to harmful meme detection. The router learns which legal tool trajectory helps a particular meme decision under a limited budget. During routing optimization, tool execution is replayed from an offline cache; policy updates neither fine-tune the LMM tools nor repeatedly query them. Training has three stages.

Stage 1: cached teacher trajectory construction. We first run CAMR and MCP on the training split to obtain $\mathcal{P}_K$ and G_x . For every node

$v \in \mathcal{V}_x$, the assigned frozen LMM tool is queried once under the node schema, and the observation $o_v = (r_v, p_v, q_v, c_v)$ is stored in $\mathcal{C}_{\text{tool}}(x, v)$. A fuller verification pass then reads observations from the cache, executes either the static DAG or all planned tools within a generous budget, and passes the resulting observations through MPRE and CBDF. For each sample, we build an approximate oracle route by choosing a low-cost subset that preserves the correct prediction, resolves MCP-marked mandatory layers, and contributes evidence later used by MPRE. When no subset gives the correct prediction, the teacher keeps the route with the best evidence coverage and calibration score. The teacher is used only for initialization and is not treated as ground-truth reasoning.

Stage 2: supervised routing warm-start. The lightweight router is first trained to imitate the oracle trajectory under the same frontier mask and STOP gate:

$$\mathcal{L}_{\text{sup}} = - \sum_t \log \pi_\phi(a_t^{\text{oracle}} \mid s_t). \quad (9)$$

This warm-start teaches basic routing behavior, such as when event grounding, image-text conflict checking, cultural decoding, or target-group analysis should be executed before stopping.

Stage 3: cached policy optimization. We refer to the router-level objective as Tool-meme-GRPO. For each labeled sample (x, y) , CAMR and MCP are run once to obtain $\mathcal{P}_K$ and a repaired DAG G_x . The old router $\pi_{\phi_{\text{old}}}$ samples J legal trajectories under the same DAG, STOP-gate, schema, and budget constraints. Each trajectory is replayed in the cached environment by reading $\mathcal{C}_{\text{tool}}(x, v)$ for selected nodes, then running MPRE and CBDF to obtain the final prediction, confidence, evidence coverage, unresolved uncertainty, and tool cost. The reward is

$$\begin{aligned} R(\Gamma_x) = & R_{\text{task}} \\ & + \lambda_{\text{ev}} R_{\text{ev}} + \lambda_{\text{cov}} R_{\text{cov}} \\ & + \lambda_{\text{cal}} R_{\text{cal}} - \lambda_{\text{cost}} C_{\text{tool}} \\ & - \lambda_{\text{unc}} U - \lambda_{\text{es}} P_{\text{early}}, \end{aligned} \quad (10)$$

where the task term is class-sensitive:

$$R_{\text{task}} = \begin{cases} +1, & \hat{y} = y, \\ -\alpha, & \hat{y} = 0, y = 1, \\ -\beta, & \hat{y} = 1, y = 0. \end{cases} \quad (11)$$

We set $\alpha \geq \beta$ so false negatives receive a larger penalty, while avoiding a policy that predicts harmful for most samples. The evidence-use reward

credits tools only when their outputs are used by MPRE:

$$R_{\text{ev}} = \frac{|\{o \in \mathcal{O}_L : o \text{ used by MPRE}\}|}{\max(1, |\mathcal{O}_L|)}. \quad (12)$$

The coverage term rewards resolving required semantic layers,

$$R_{\text{cov}} = \frac{|\text{resolved}(\mathcal{M}_x)|}{\max(1, |\mathcal{M}_x|)}, \quad (13)$$

where $\mathcal{M}_x$ is the set of mandatory layers assigned by MCP. Cost is normalized by the budget,

$$C_{\text{tool}} = \frac{\sum_{v \in \mathcal{H}_L} c_v}{B}, \quad (14)$$

and calibration is measured by the agreement between final confidence and correctness, e.g.,

$$R_{\text{cal}} = -|c_{\text{final}} - \mathbb{I}(\hat{y} = y)|. \quad (15)$$

Unresolved uncertainty is computed from the final calibrated confidence:

$$U = 2(1 - c_{\text{final}}), \quad (16)$$

where $c_{\text{final}} \in [0.5, 1]$, so $U \in [0, 1]$. This term lightly penalizes decisions that remain close to the decision boundary after evidence synthesis. The early-stop penalty is

$$P_{\text{early}} = \mathbb{I}[\text{STOP} \wedge \mathcal{M}_L \neq \emptyset](1 + \eta_{\text{es}} \mathbb{I}[\hat{y} \neq y]), \quad (17)$$

so premature stopping receives a larger penalty when it causes an incorrect final prediction. Schema validity is handled by the action mask and output validation; malformed outputs become unresolved evidence and are penalized through R_{cov} , U , and P_{early} . High-reward routes are those that make the right prediction, use evidence later consumed by MPRE, cover mandatory layers, and avoid unnecessary tool cost.

Rewards are normalized within the sampled trajectory group for the same meme,

$$A_j = \frac{R_j - \text{mean}(\{R_i\}_{i=1}^J)}{\text{std}(\{R_i\}_{i=1}^J) + \epsilon}, \quad (18)$$

so the policy compares routes for the same instance rather than rewards from unrelated samples. We define the clipped term and objective as

$$\begin{aligned} \ell_j &= \min(r_j A_j, \text{clip}(r_j, 1 - \epsilon_c, 1 + \epsilon_c) A_j), \\ \mathcal{L}_{\text{ATR}} &= -\frac{1}{J} \sum_{j=1}^J \ell_j, \end{aligned} \quad (19)$$

Algorithm 2 RL-Guided Training for ATR

Require: Labeled samples $\mathcal{D}$, memory bank $\mathcal{K}$, initialized router π_ϕ , group size J , clipping coefficient ϵ_c

Ensure: Trained routing policy π_ϕ

- 1: Run CAMR and MCP, query each planned frozen tool once, and cache node observations
 - 2: Execute fuller teacher routes from the cache and construct approximate oracle trajectories
 - 3: Warm-start π_ϕ with supervised routing loss under the DAG mask and STOP gate
 - 4: **for** each minibatch from $\mathcal{D}$ **do**
 - 5: **for** each sample (x, y) **do**
 - 6: Load cached $\mathcal{P}_K$, repaired DAG G_x , and node observations
 - 7: Sample J legal trajectories $\{\Gamma_x^{(j)}\}_{j=1}^J$ from $\pi_{\phi_{\text{old}}}$ under the frontier mask
 - 8: **for** each trajectory $\Gamma_x^{(j)}$ **do**
 - 9: Replay selected node observations under G_x , budget B , schema constraints, and STOP gate
 - 10: Run MPRE and CBDF to obtain $(\hat{y}_j, c_j, E_j)$
 - 11: Compute R_j from task correctness, evidence use, coverage, calibration, cost, uncertainty, and early stopping
 - 12: **end for**
 - 13: Normalize $\{R_j\}_{j=1}^J$ into advantages $\{A_j\}_{j=1}^J$
 - 14: Accumulate clipped group-relative policy loss
 - 15: **end for**
 - 16: Update ϕ and refresh $\pi_{\phi_{\text{old}}}$ after the update window
 - 17: **end for**
 - 18: **return** π_ϕ
-

where $r_j = \pi_\phi(\Gamma_x^{(j)}) / \pi_{\phi_{\text{old}}}(\Gamma_x^{(j)})$. The clipping term limits abrupt router updates while still favoring trajectories that improve the decision with fewer or better-selected tools.

B Detailed Experimental Settings

B.1 Datasets

Table B.1 gives the test-set composition used in our experiments. FHM is the Facebook Hateful Memes benchmark, where the label depends on the joint meaning of an image and its overlaid text (Kiel et al., 2020). HarM follows the HarMeme/Harm-C line and covers COVID-19 and socio-political memes; as in prior work, we merge partially harmful and very harmful examples into the positive class (Pramanick et al., 2021a; Sharma et al., 2022b; Liu et al., 2025). MAMI focuses on misogynistic memes targeting women and provides both binary and fine-grained annotations (Fersini et al., 2022). Because Tool-meme’s decision module is trained for harmful/harmless prediction, all main experiments use the binary setting.

B.2 Baselines

We compare Tool-meme with closed-source direct LMMs, open-source direct LMMs, and closed-source agent-based harmful meme detectors. The

Datasets	Test	
	#harmful	#harmless
HarM	124	230
FHM	250	250
MAMI	500	500

Table B.1: Statistics of test sets.

direct-model baselines test whether broad vision-language pretraining is sufficient without task-specific evidence acquisition. The agent-based baselines place intermediate evidence, debate, or social reasoning between the raw meme and the final decision, which makes them closer comparisons to Tool-meme.

GPT-4o is used as a closed-source direct LMM baseline with the meme image and text as input. It predicts harmfulness without retrieval, explicit planning, or external tool execution, so its result reflects the behavior of a general multimodal model under a direct decision prompt.

Gemini-1.5-Flash is included as an efficient proprietary multimodal model from the Gemini 1.5 family. We evaluate it in the same direct-prediction setting as GPT-4o, which helps check whether speed-oriented multimodal pretraining alone can handle implicit harmfulness cues in memes.

Claude-3.5-Sonnet provides another closed-source vision-language comparison with a different model family and decoding behavior. It receives no retrieved context or planned evidence chain, allowing us to separate the effect of Tool-meme’s evidence acquisition from the capacity of a capable general LMM.

LLaVA-1.6-34B is an open-source visual instruction-following LMM from the LLaVA-NeXT family (Liu et al., 2024). We use it as a large open-source direct-prediction baseline to test whether scaling an open visual instruction model can close the gap without retrieval or tool routing.

Qwen2.5-VL-72B is a large open-source vision-language model with stronger visual grounding and document/image understanding capabilities (Bai et al., 2025b). It is evaluated in the same direct-prediction setting to compare Tool-meme against a high-capacity open LMM.

Qwen3-VL-32B-Instruct is an instruction-tuned open-source vision-language model from the Qwen3-VL family (Bai et al., 2025a). It provides an additional open LMM comparison with a different model family and scale from LLaVA and Qwen2.5-VL.

MIND is a zero-shot multi-agent framework for harmful meme detection (Liu et al., 2025). It retrieves similar memes from an unlabeled reference set, derives bidirectional insights from the retrieved neighbors, and uses multi-agent debate to arbitrate the final harmfulness decision.

Ask-Acquire-Understand is an agent-based framework for social abuse detection in memes (Lin et al., 2025). It asks targeted questions about the image, acquires answers from an LMM-based vision expert, and uses the resulting descriptions to support zero-shot meme understanding.

ExplainHM++ detects harmful memes through retrieval-augmented debate between large multimodal models (Lin et al., 2026). It builds harmful and harmless rationales, injects retrieved meme examples into the debate, and uses a judge model to combine the rationales with multimodal content.

B.3 Implementation Details

Unless otherwise specified, Tool-meme uses CLIP ViT-L/14 as a frozen multimodal encoder and the 10,000,000-record SCM-10M-v1 FAISS index as the CAMR retrieval database (Radford et al., 2021; Johnson et al., 2021). The LMM tools are frozen API executors. GPT-4o is the default executor, and Claude-3.5-Sonnet and Gemini-1.5-Flash are used in the compatibility study. No LMM weights are updated; training is limited to the lightweight masked ATR router.

For retrieval and planning, we use $K = 10$ neighbors and limit the MCP-generated JSON-constrained DAG to depth 4. Before optimizing the router, we cache the output of every executable DAG node for each training sample and replay trajectories against this cache. ATR uses a tool budget $B = 6$, a short-circuit threshold $\tau_{sc} = 0.80$, learning rate 3×10^{-5} , clipping coefficient 0.2, and group size $J = 8$. The reward and decision hyperparameters are fixed as $\lambda_{ev} = 0.20$, $\lambda_{cov} = 0.30$, $\lambda_{cal} = 0.20$, $\lambda_{cost} = 0.15$, $\lambda_{unc} = 0.15$, $\lambda_{es} = 0.50$, $\alpha = 1.25$, $\beta = 1.00$, $\eta_{es} = 1.00$, $\tau_{stop} = 0.80$, $\tau_{conflict} = 0.30$, and $\tau_{dec} = 0.70$. We selected these values on the development split with a small coarse grid and kept them unchanged across datasets and backbones.

All LMM baselines use temperature-zero decoding during evaluation unless the original baseline protocol requires sampling. When settings are comparable, we keep data splits, prompt budgets, and retrieval pools aligned across methods. Repeated runs are reported as averages.

Retrieval / routing source	Target	GPT-4o		Tool-meme		
		Acc. $\uparrow$	M-F1 $\uparrow$	Acc. $\uparrow$	M-F1 $\uparrow$	Calls $\downarrow$
FHM+MAMI	HarM	67.51	60.29	71.93	70.54	3.78
HarM+MAMI	FHM	68.80	68.25	73.21	72.36	3.91
FHM+HarM	MAMI	81.00	80.82	84.42	83.17	3.54
Average	–	72.44	69.79	76.52	75.36	3.74

Table C.1: Cross-dataset generalization. Tool-meme is trained with retrieval and routing sources from the other two datasets, and evaluated on the target dataset.

All router-training experiments are conducted on a single NVIDIA A100 GPU with 80 GB of memory, 32 CPU cores, 128 GB of RAM, and NVMe storage. Because the LMM-based tools are frozen API executors, GPU memory is mainly used for CLIP feature extraction, lightweight router optimization, and local minibatch replay of cached trajectories. The offline cache construction stage is the most time-consuming part because it queries the frozen LMM tools for the planned DAG nodes. After the cache is built, both supervised router warm-start and cached policy optimization run on the single GPU. Full API-based Tool-meme evaluation is executed on the evaluation splits, with wall-clock time depending on API latency and rate limits. For this reason, we report the cached policy-optimization reward curve and run complete API-based evaluation only for the final selected checkpoints and routing-policy comparisons.

C Supplementary Experimental Analysis

C.1 Cross-Dataset Generalization

The main experiments train, retrieve, and evaluate within the same benchmark. We add a leave-one-dataset-out setting to check how much of Tool-meme’s behavior carries over when the retrieval pool and router warm-start traces come from different meme domains. For each target dataset, CAMR and ATR use only the other two datasets during memory-bank construction and routing optimization; the target training and development splits are not used. At inference time, the target image–text pair can retrieve context only from this source-domain memory bank.

Table C.1 reports a drop from the fully in-domain setting but retains a clear margin over direct GPT-4o. Tool-meme improves average Macro-F1 by 5.57 points over direct GPT-4o, with 3.74 tool calls per sample on average. The MAMI gain is smaller than the FHM gain, suggesting that misogyny-specific cues transfer less cleanly from hate and public-health/political memes. The result is still

useful: the learned router does not depend entirely on dataset-local neighbors, and the same operations of grounding, conflict checking, and rhetorical interpretation remain helpful after the retrieval domain changes.

C.2 Adaptive Routing Analysis

Table C.2 isolates the routing policy. Random routing loses accuracy because it may execute tools before their required evidence is available. The heuristic variant respects a fixed order and is more stable, but it still cannot react to sample-level uncertainty. Static DAG execution follows the MCP plan and runs all feasible nodes, whereas all-tools execution ignores both the plan and the difficulty of the current meme. This comparison matches the agent-efficiency problem studied in prior tool-use work: an extra call is useful only when it changes the evidence available to the decision module (Schick et al., 2023; Wang et al., 2025).

RL-guided ATR gives the best trade-off in this table: it improves Macro-F1 while using fewer calls and lower latency than the non-learned policies. For the three deployment-audited policies, it also has the lowest measured relative API cost.

Routing	M-F1 $\uparrow$	Calls $\downarrow$	Lat. $\downarrow$	Cost $\downarrow$
Random	71.37	4.92	18.62	–
Heuristic	73.38	4.57	16.94	–
DAG	74.97	5.93	21.46	0.776
All-tools	75.26	8.00	38.41	1.000
RL-ATR	78.70	3.64	9.87	0.485

Table C.2: Routing-policy analysis. Latency is measured in seconds. Relative API cost is normalized to all-tools execution and reported only for the policies included in the deployment audit.

Figure 5 shows the cached policy-optimization signal used to train ATR. Each point is the average normalized trajectory reward for legal routes sampled in a mini-batch. The LMM tools remain frozen, and their node-level observations are pre-computed; the router only learns from replayed tra-

jectories under DAG masks and STOP gates. The raw curve fluctuates because each update samples a different mini-batch and a different set of legal routes. Its moving average rises early and then levels off, indicating that the router increasingly favors routes that preserve the correct decision, cover required layers, use evidence consumed by MPRE, and avoid unnecessary calls.

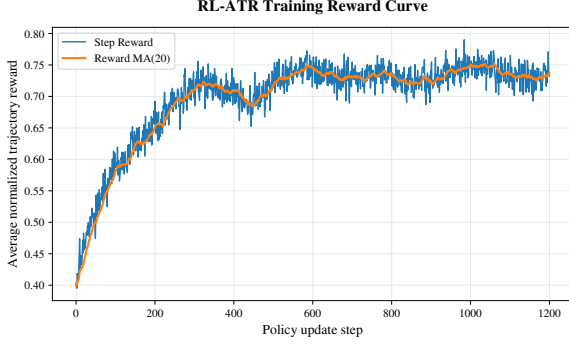


Figure 5: Training reward curve for cached RL-guided ATR optimization.

D Memory Bank Construction

The memory bank $\mathcal{K}$ is used for socio-cultural context rather than as a source of test answers. It is built from training/development splits, public template encyclopedia entries, public social-platform posts, and permitted external image-text samples. Test samples and their near-duplicate clusters are excluded from retrieval construction.

Each candidate stores the image or image embedding, OCR/title/post text, image description, weak label distribution, event clues, cultural/template tags, time, source, anonymized source identifier, and reliability score. Exact duplicates are removed with image and text hashing, and near-duplicate templates are compressed using joint perceptual-hash and OCR-masked visual/text similarity. At inference time, CAMR retrieves only candidates earlier than the target sample and outside the same test split or near-duplicate cluster. Conflicting labels are kept as distributions with conflict flags, and reliability is passed to later modules through κ_K .

Memory-bank scale and composition. To make the 10M memory-bank setting auditable, we additionally evaluate nested memory-bank subsets and an external-only memory variant. All subsets are sampled from the same de-duplicated candidate pool, and all test samples and near-duplicate clusters remain excluded. The 10M setting corresponds

to the full Tool-meme(GPT-4o) result reported in Table 1.

Table D.1 and Table D.2 show a consistent but saturating trend. Compared with the no-memory variant, a 100K memory bank already improves the average M-F1 by 0.60 points, suggesting that even a relatively small retrieval corpus can provide useful template-level context. Increasing the memory size to 1M and 3M further improves the average M-F1 by 1.03 and 1.27 points, respectively. The full 10M memory bank achieves the best performance, with a 1.45-point improvement over the no-memory variant. However, the gain from 3M to 10M is only 0.18 points, indicating diminishing returns rather than reliance on a small number of leaked or memorized examples. Table D.3 further shows that the external-only memory already contributes most of the gain, while the full memory adds only 0.26 average M-F1 points, supporting that the memory bank primarily provides public context, template cues, and rhetorical structures rather than benchmark answers.

E Complexity and Cost Analysis

The inference cost of Tool-meme includes encoding, retrieval, planning, selected tool execution, evidence synthesis, and decision fusion:

$$C_{\text{Tool-meme}} = C_{\text{enc}} + C_{\text{ret}}(K) + C_{\text{MCP}} + \sum_{v \in \mathcal{H}} C_{\tau(v)} + C_{\text{MPRE}} + C_{\text{CBDF}}. \quad (20)$$

The dominant variable term is the selected tool cost. A static all-tools workflow pays $\sum_{j=1}^8 C_{\tau_j}$ for every sample, whereas ATR executes only nodes retained by the repaired MCP plan and the budget:

$$|\mathcal{H}| \leq \min(|\mathcal{V}_x|, B, 8). \quad (21)$$

The expected saving relative to running every tool is

$$\Delta C = \sum_{j=1}^8 C_{\tau_j} - \mathbb{E}_x \left[\sum_{v \in \mathcal{H}_x} C_{\tau(v)} \right]. \quad (22)$$

The DAG frontier also supports parallel execution. If $\{\mathcal{B}_r\}_{r=1}^{R_x}$ are topological execution batches, the routing latency is approximated by

$$L_{\text{ATR}} \approx \sum_{r=1}^{R_x} \max_{v \in \mathcal{B}_r} L_{\tau(v)}. \quad (23)$$

This routing rule removes redundant calls on simple samples while keeping additional computation available for samples with incomplete or conflicting evidence.

Memory size	HarM-Acc	HarM-M-F1	FHM-Acc	FHM-M-F1	MAMI-Acc	MAMI-M-F1	Avg.
0	72.53	71.29	76.90	74.77	85.93	85.69	77.25
100K	72.85	71.56	77.20	76.08	86.12	85.92	77.85
1M	73.37	72.02	77.64	76.51	86.50	86.31	78.28
3M	73.51	72.25	77.86	76.74	86.62	86.56	78.52
10M	73.64	72.37	78.03	76.98	86.81	86.74	78.70

Table D.1: Effect of memory-bank scale. The 10M row corresponds to the full Tool-meme(GPT-4o) setting. Smaller memory banks are nested subsets of the same de-duplicated candidate pool. The average score is computed over the main metric, M-F1, of each dataset.

Memory	HarM	FHM	MAMI	Avg.
0	0.00	0.00	0.00	0.00
100K	+0.27	+1.31	+0.23	+0.60
1M	+0.73	+1.74	+0.62	+1.03
3M	+0.96	+1.97	+0.87	+1.27
10M	+1.08	+2.21	+1.05	+1.45

Table D.2: Effect of memory-bank scale. Values are M-F1 improvements over the no-memory variant. The improvement increases with memory size but exhibits diminishing returns.

F Human Evaluation

Automatic classification metrics do not show whether an evidence chain is useful for inspecting a harmful-meme decision. We run a human evaluation of explanation quality to measure this aspect directly. The evaluation uses 180 test examples, with 60 examples from each of HarM, FHM, and MAMI. The sample is label-balanced and includes both easy cases and cases where at least one system is wrong. For each example, we collect evidence chains from GPT-4o and Tool-meme. System names are anonymized and evidence-chain order is randomized.

Three annotators, including Ph.D. students in computer science and media studies, independently score each evidence chain under the same guideline. The guideline contains five dimensions: Evidence Relevance (ER), Evidence Completeness (EC), Cross-Modal Faithfulness (CMF), Cultural/Contextual Grounding (CCG), and Decision Support (DS). Each dimension is scored from 1 to 5, where 1 indicates that the evidence is irrelevant or misleading, and 5 indicates that the evidence is specific, faithful to the meme, and sufficient for checking the decision. After mapping scores into low, medium, and high quality bins, the inter-annotator agreement is substantial, with Fleiss’ $\kappa = 0.72$.

System	ER	EC	CMF	CCG	DS	Avg.
GPT-4o	3.48	3.12	3.34	2.81	3.20	3.19
Tool-meme	4.31	4.12	4.24	4.06	4.18	4.18

Table F.1: Human evaluation of evidence-chain quality. Scores are averaged over 180 examples and three annotators.

Table F.1 shows higher Tool-meme scores on all five dimensions. The largest margins are in Cultural/Contextual Grounding and Evidence Completeness, which are the two dimensions most affected by CAMR retrieval and MCP planning. Annotators often found that direct GPT-4o explanations mention visible offensive terms but leave the cultural or rhetorical link implicit. Tool-meme exposes the retrieved context, selected tools, local judgments, and consensus evidence, so annotators can more easily check where the final decision comes from. A paired Wilcoxon signed-rank test gives a significant difference between Tool-meme and GPT-4o on the average human score ($p < 0.01$).

Using the same anonymized evidence chains, we also test whether LMM-based scoring can approximate human quality judgments. GPT-4o-mini and GPT-4o score the 180 chains with the prompt below, and we compute Pearson correlations against the averaged human scores. Table F.2 shows that GPT-4o has the higher average correlation, while GPT-4o-mini remains directionally consistent. We use these scores only as an auxiliary signal; the human ratings in Table F.1 remain the main evidence for interpretability.

Variant	HarM-Acc	HarM-M-F1	FHM-Acc	FHM-M-F1	MAMI-Acc	MAMI-M-F1	Avg.
Tool-meme w/o memory	72.53	71.29	76.94	74.72	85.93	85.62	77.21
Tool-meme + external-only memory	73.38	72.16	77.74	76.69	86.51	86.47	78.44
Tool-meme + full memory	73.64	72.37	78.03	76.98	86.81	86.74	78.70

Table D.3: Effect of external memory composition. The external-only memory excludes all benchmark train/dev/test instances. The full memory uses the complete de-duplicated memory bank, where benchmark train/dev labels are removed and all test duplicates are excluded.

Dimension	GPT-4o-mini	GPT-4o
ER	0.704	0.763
EC	0.681	0.741
CMF	0.697	0.752
CCG	0.668	0.729
DS	0.715	0.758
Average	0.693	0.749

Table F.2: Correlation between LMM-based and human evaluations.

The prompt used is as follows:

I have a multimodal meme, a candidate harmfulness prediction, and an evidence chain generated by a detection system. I want to evaluate how useful and faithful the evidence chain is for checking the decision based on the following criteria:

1. Evidence Relevance: Does the evidence directly address the harmfulness decision rather than discussing unrelated visual or textual details?
2. Evidence Completeness: Does the evidence cover the necessary semantic layers, such as target group, harmful implication, event context, and unresolved uncertainty?
3. Cross-Modal Faithfulness: Is the evidence faithful to the joint meaning of the image, overlaid text, OCR, and caption?
4. Cultural/Contextual Grounding: Does the evidence correctly use retrieved socio-cultural context, meme templates, political or public-event background, and community-specific symbols?
5. Decision Support: Does the evidence chain explain why the final prediction should be harmful or harmless, and does it identify cases that need human review?

Task: Please evaluate the following evidence chain on each criterion and provide a score from 1 to 5, where 1 means the evidence is irrelevant or misleading and 5 means the evidence is specific, faithful, and sufficient for checking the decision.

Meme text: [text]

Image/OCR evidence: [image evidence]

Retrieved context: [retrieved context]

System prediction: [harmful or harmless]

Evidence chain to evaluate: [evidence chain]

Please provide the result as JSON with scores for ER, EC, CMF, CCG, and DS, plus one brief justification for each score.

G Additional Transfer and Control Experiments

This section reports additional evidence that complements the main three-dataset comparison. It separates unlike training regimes, evaluates transfer

beyond the original binary English-centric setting, and introduces controls that distinguish retrieval from routing. Unless noted otherwise, the LMM executors remain frozen and only the lightweight ATR router is trained.

G.1 HarM Task-Specific References and Claim Scope

Table G.1 places the frozen-executor results beside stronger task-specific HarM systems. MO-MENTA, MaskPrompt, and MR.HARM use task-specific supervised training, prompt tuning, or reasoning distillation, so they are reference systems rather than like-for-like baselines. Their reported absolute HarM scores are higher than Tool-meme’s. The matched claim supported by our experiments is therefore narrower: Tool-meme outperforms the evaluated direct-LMM and agentic baselines while keeping the LMM executors frozen and training only the ATR router.

G.2 Multilingual and Multicultural Transfer

Multi³Hate contains parallel memes in English, German, Spanish, Hindi, and Mandarin with culturally localized annotations from the United States, Germany, Mexico, India, and China, respectively (Bui et al., 2025). We use it as a transfer diagnostic: the router trained for the original binary task is transferred without updating its weights, and the binary decision schema is retained. Each table cell reports Accuracy / Macro-F1.

Tool-meme improves average Macro-F1 by 4.43 points over direct GPT-4o and by 5.10 points over direct Qwen2.5-VL-72B. It also raises the worst-locales accuracy from 64.16 to 70.04 relative to GPT-4o, with positive gains in all five evaluated locales. These results provide evidence for the tested languages and cultural annotation settings; they do not establish universal multilingual or real-time coverage.

G.3 Fine-Grained MAMI Transfer

The main experiments use the binary protocols of HarM, FHM, and MAMI Subtask A. We addition-

Method	Training regime	Acc.	M-F1	Comparison role
MOMENTA (Pramanick et al., 2021b)	Task-specific supervised training	83.82	82.80	Reported task-specific reference
MaskPrompt (Cao et al., 2022; Lin et al., 2023)	Task-specific prompt tuning	84.47	81.51	Reported task-specific reference
MR.HARM (Lin et al., 2023)	Two-stage small-model fine-tuning and reasoning distillation	86.16	85.43	Reported task-specific reference
ExplainHM++ (Lin et al., 2026)	Agentic baseline rerun in our protocol	70.84	69.50	Like-for-like agentic comparison
Tool-meme	Frozen LMM executors; lightweight ATR router trained	73.64	72.37	Evaluated setting in this work

Table G.1: HarM reference results under different training regimes. The task-specific systems are not matched to Tool-meme in trainable parameters or inference protocol and are not used for a direct comparison. The MaskPrompt score pair follows the comparison reported by Lin et al. (2023); Cao et al. (2022) introduced the underlying prompting method.

Method	en-US	de-DE	es-MX	hi-IN	zh-CN	Average	Worst Acc.
Direct Qwen2.5-VL-72B	75.43 / 74.95	69.82 / 69.20	69.53 / 69.19	65.57 / 64.62	68.01 / 67.93	69.67 / 69.18	65.57
Direct GPT-4o	75.64 / 75.34	71.54 / 71.07	70.67 / 70.26	64.16 / 63.46	69.85 / 69.10	70.37 / 69.85	64.16
Tool-meme (GPT-4o)	78.76 / 78.51	75.32 / 74.98	74.83 / 74.43	70.04 / 69.45	74.59 / 74.01	74.71 / 74.28	70.04

Table G.2: Frozen-router transfer to Multi³Hate. Each locale column reports Accuracy / Macro-F1 (%); Average is the unweighted mean across the five displayed locale scores.

Label	Support	Acc.	F1	Dev threshold
Shaming	146	75.07	73.28	0.47
Stereotype	350	84.19	82.83	0.52
Objectification	348	81.86	80.52	0.50
Violence	153	72.82	68.94	0.44

Table G.3: Label-wise Tool-meme results and development thresholds on MAMI Subtask B. The unweighted mean of the four displayed label F1 scores is 76.39.

ally use MAMI Subtask B as a task-granularity diagnostic (Fersini et al., 2022). The binary-trained router remains frozen, while the final decision schema is changed to four independent probabilities for shaming, stereotype, objectification, and violence. Thresholds are selected on the development split and then held fixed on the test set. We report the label-wise Tool-meme run under the paper’s default frozen-executor setting and use it only to document behavior beyond the binary output space; the matched routing comparison is evaluated separately in Section G.4.

The recorded Tool-meme run attains label-wise F1 scores from 68.94 to 82.83 and an unweighted four-label mean of 76.39. This result documents extension beyond the binary output space; it is not used for a matched-executor or causal routing claim. We use Table G.5 for the controlled routing comparison.

G.4 Disentangling CAMR and Adaptive Routing

We first isolate the contribution of retrieval without executing any tools. CAMR-context-only supplies retrieved image descriptions and cultural/event context but excludes neighbor labels and the calibrated prior ρ_K . CAMR-full adds both signals. These controls are distinct from the “w/o CAMR” row in Table 2: CAMR-only removes the downstream tool pipeline, whereas w/o CAMR retains the remaining modules and removes retrieval.

Retrieved descriptions and cultural/event context contribute 2.52 Macro-F1 points over direct GPT-4o. Adding the calibrated neighbor-label prior contributes a further 0.62 points. We then hold CAMR-full, the GPT-4o executor, data splits, tool library, applicable MCP outputs, and final decision protocol fixed, changing only the routing mechanism.

RL-ATR improves average Macro-F1 by 2.58 points over the supervised router while reducing the average selected-tool count from 4.11 to 3.64. It exceeds the global fixed chain by 4.77 points at nearly the same call count, and exceeds all-tools by 3.44 points with 54.5% fewer selected tool calls. We use this fixed-CAMR policy sweep, rather than the component-removal row “w/o RL-ATR” in Table 2, to isolate routing-policy effects.

System	Retrieved context	Neighbor labels / ρ_K	Avg. M-F1	Gain over direct
Direct GPT-4o	×	×	69.79	—
CAMR-context-only	✓	×	72.31	+2.52
CAMR-full	✓	✓	72.93	+3.14

Table G.4: Controlled contribution of CAMR. Both CAMR variants make a direct decision with zero tool calls.

Routing policy with CAMR-full	Mechanism	HarM	FHM	MAMI	Avg. M-F1	Calls
Global fixed chain	Same subset and order for every meme	67.75	72.20	81.85	73.93	3.67
Static MCP-DAG	Execute every feasible MCP node	68.75	73.16	83.00	74.97	5.93
All-tools	Execute all eight tools	69.20	74.00	82.58	75.26	8.00
SL router	Supervised warm-start only	70.05	74.42	83.89	76.12	4.11
RL-ATR	Reinforcement-learned adaptive routing	72.37	76.98	86.74	78.70	3.64

Table G.5: Routing-policy control with CAMR-full fixed. Dataset columns report Macro-F1 (%); Calls is the average number of selected tools per sample.

G.5 Reward-Design Diagnostics

We test whether the router can obtain in-domain accuracy by exploiting dataset-local label priors or cached output combinations. The label-only objective retains task correctness, label+cost adds only the tool-cost penalty, and the full reward uses all terms defined in Appendix Section A.2. Leave-one-dataset-out (LODO) evaluation uses only the other two datasets for retrieval and routing sources.

Label-only nearly matches the full reward in-domain but loses 2.51 Macro-F1 points under LODO and has a 2.01-point larger ID-LODO gap. Adding cost alone produces the fewest calls but the lowest semantic-layer coverage and the highest premature-stop error. The full reward yields the best cross-domain result, calibration, coverage, and prior-shuffle stability, supporting the role of the semantic, calibration, and stopping terms beyond label correctness.

G.6 End-to-End Portability to Open-Source Executors

We run the complete Tool-meme pipeline with three open-source executors in addition to evaluating them as direct baselines. Each direct/Tool-meme pair uses the same checkpoint, image preprocessing, and decoding; only the native chat template differs across model families.

All open-source executors use the same SGLang/BF16 configuration, a 4,096-token context, standardized image preprocessing, greedy decoding, and a shared JSON schema ($K = 10$, $B = 6$, and $\tau_{sc} = 0.80$); direct baselines use matched checkpoints, prompts, and decoding settings.

The gains are positive for all three executors and

all three datasets. They support portability across the evaluated model families; they do not imply that parameter count alone explains performance or that every future executor will benefit equally.

H Empirical Deployment Audit

Appendix Section E gives the analytic decomposition. For the empirical audit, we instrument the complete pipeline on July 12, 2026, with the pinned gpt-4o-2024-11-20 snapshot and recorded rates of USD 2.50 per million input tokens, USD 1.25 per million cached input tokens, and USD 10.00 per million output tokens; cached-input use is zero.

The audit results below show that, relative to all-tools execution, RL-ATR reduces API cost by 51.5% and mean end-to-end latency by 74.3%, while improving average Macro-F1 from 75.26 to 78.70. The measurements are tied to the stated snapshot, date, image size, and pricing assumptions; later API prices or serving conditions require recomputation.

Runs use a warm model/index, batch size one, parallel DAG-frontier execution, high-detail images, and neither fast exit nor full-response caching. A 1024×768 image contributes $85 + 4 \times 170 = 765$ input tokens. Images go to MCP, fast judgment, and selected tools; later stages are text-only.

Reward	Coverage	ECE ↓	Early-stop err. ↓	Prior-shuffle Δ ↑
Label only	78.6	8.7	5.9	-4.8
Label + cost	64.3	10.1	13.7	-5.6
Full reward	89.1	4.9	4.6	-1.9

Table G.7: Behavioral reward diagnostics (%). Prior-shuffle Δ is the Macro-F1 change after shuffling retrieved label priors; a value closer to zero indicates lower sensitivity.

Reward	ID Avg.	LODO HarM	LODO FHM	LODO MAMI	LODO Avg.	ID-LODO gap	Calls
Label only	78.20	68.90	69.95	79.70	72.85	5.35	5.28
Label + cost	77.65	68.10	69.20	78.40	71.90	5.75	2.76
Full reward	78.70	70.54	72.36	83.17	75.36	3.34	3.74

Table G.6: Reward ablation under in-domain (ID) and leave-one-dataset-out evaluation. Performance values are average or dataset Macro-F1 (%). Calls is measured in the LODO setting, explaining why the full-reward value (3.74) differs from the in-domain routing value (3.64).

Executor and setting	HarM Acc.	HarM M-F1	FHM Acc.	FHM M-F1	MAMI Acc.	MAMI M-F1	Avg. M-F1 / Calls
LLaVA-1.6-34B, direct	67.51	61.59	64.00	63.51	71.30	71.28	65.46 / 0
Tool-meme (LLaVA-1.6-34B)	71.02	69.34	72.21	71.47	81.86	81.43	74.08 / 3.82
Qwen2.5-VL-72B, direct	69.37	63.18	71.27	71.02	78.66	77.52	70.57 / 0
Tool-meme (Qwen2.5-VL-72B)	74.94	71.58	77.63	76.02	84.07	83.86	77.15 / 3.66
Qwen3-VL-32B-Instruct, direct	66.20	65.67	67.65	67.32	74.76	72.25	68.41 / 0
Tool-meme (Qwen3-VL-32B-Instruct)	72.16	70.85	74.21	73.63	83.74	83.21	75.89 / 3.73

Table G.8: End-to-end Tool-meme results with open-source executors. Scores are percentages; Calls is the average number of selected tools. Relative to its matched direct baseline, Tool-meme improves average Macro-F1 by 8.62, 6.58, and 7.48 points for LLaVA, Qwen2.5-VL, and Qwen3-VL-Instruct, respectively.

RL-ATR stage	Calls/sample	Image	Input/call	Output/call	Aggregate input	Aggregate output
MCP	1.00	High	2,245	310	2,245.0	310.0
Fast judgment	1.00	High	1,245	55	1,245.0	55.0
Selected tools	3.64	High	1,905	210	6,934.2	764.4
MPRE	1.00	No	2,075	330	2,075.0	330.0
CBDF	1.00	No	950	130	950.0	130.0
LMM schema repair	0.08	No	430	80	34.4	6.4
Total	7.72	–	–	–	13,483.6	1,595.8

Table H.1: Per-stage request and token accounting for RL-ATR. Input/output columns are tokens.

Policy	Selected tools	LMM requests	Input	Cached input	Output	USD/sample	Relative cost
RL-ATR	3.64	7.72	13,484	0	1,596	0.0497	0.485
Static MCP-DAG	5.93	10.05	22,760	0	2,260	0.0795	0.776
All-tools	8.00	12.16	30,100	0	2,720	0.1025	1.000

Table H.2: Empirical API-cost audit. Relative cost is normalized to all-tools execution.

Policy	RPC sum	CAMR	MCP	Fast	Tool critical path	MPRE	CBDF	Queue/retry	E2E mean
RL-ATR	18.72	0.18	1.36	0.82	4.94	1.22	1.08	0.27	9.87
Static MCP-DAG	31.84	0.18	1.36	0.82	14.52	2.05	1.31	1.22	21.46
All-tools	44.96	0.18	1.36	0.82	29.84	2.72	1.43	2.06	38.41

Table H.3: Latency audit in seconds. RPC sum adds all LMM request service times even when requests overlap; Tool critical path is the actual wait within the parallel tool stage; E2E mean is wall-clock latency from sample receipt to final JSON.

I Prompt Set

This section reports the prompt contracts used by Tool-meme. The MCP, MPRE, and CBDF prompts share the same interface as the main-text examples and are specialized to the harmful/harmless label space used in our experiments.

MCP planning prompt

Based on the retrieved similar-meme context (event: [event], cultural tags: [tags]), diagnose the core cognitive needs of the current meme. Step 1: Identify the key semantic layers. For example, determine whether the meme contains irony, whether it depends on a public event, whether the image and text conflict, whether it invokes a cultural template, and whether it targets a protected or vulnerable group.

Step 2: Assign one tool to each semantic layer and define the execution order as DAG edges. The order should follow cognitive dependencies, such as event verification before inconsistency detection, and rhetorical scanning before culture-specific interpretation when the rhetoric depends on a meme template.

Output JSON:

```
{ "semantic_layers": ["temporal_context",
"visual_textual_alignment",
"rhetorical_intent"], "tool_dag": [ {"node":
"n1", "tool": "knowledge_grounding", "deps":
[], "priority": 1}, {"node": "n2", "tool":
"cross_modal_aligner", "deps": ["n1"],
"priority": 2}, {"node": "n3", "tool":
"rhetorical_scanner", "deps": ["n2"], "priority":
3} ] }
```

Do not add a semantic layer if it is not supported by the meme, the OCR/caption, or the retrieved context.

Tool prompt design

Shared interface. Each tool receives [query text], [image evidence], [retrieved context], [assigned semantic layer], and [previous tool outputs]. Each tool must inspect only its assigned layer and return JSON with name, trace, pred, conf, evidence, and status. The pred field must be harmful, harmless, or unknown. The status field must be valid, abstain, or failed.

Semantic Dissector: Analyze the literal and implied meaning of the text, including slang, puns, coded wording, and target references.

Rhetorical Scanner: Identify irony, sarcasm, exaggeration, metaphor, rhetorical reversal, and cases where a seemingly harmless surface meaning implies harm.

Knowledge Grounding: Verify whether the meme is consistent with retrieved event context, public figures, time-sensitive facts, and social background.

Expectation Deviator: Estimate whether the image and text intentionally violate normal expectations, such as a formal image paired with degrading, vulgar, or hostile text.

Cross-Modal Aligner: Measure visual-textual consistency using image-text similarity, visible entities, OCR, and LMM verification.

Inconsistency Amplifier: Locate contradictions across image, text, retrieval results, and previous local judgments, then determine whether the contradiction strengthens harmful intent.

Cultural Decoder: Interpret meme templates, subcultural symbols, political references, community-specific slang, and visual archetypes such as Wojak-style symbols.

Evidence Comparator: Compare the target meme with retrieved neighbors, examine label-prior shift, and identify whether retrieved examples provide reliable harmful or harmless evidence.

Execution rule. Follow the DAG generated by MCP. Start from nodes with no dependencies. Pass predecessor outputs to successor tools. If a tool confidence is above 0.80 and the remaining low-priority tools would only repeat the same evidence, ATR may short-circuit them. If a required dependency fails, mark the downstream layer unresolved rather than inventing evidence.

MPRE synthesis prompt

Synthesize the reasoning traces from the selected tools and mark evidence strength.
[Tool 1]: [trace 1] (confidence: [score 1])
[Tool 2]: [trace 2] (confidence: [score 2])
...
Step 1: Identify consensus evidence, such as multiple tools pointing to image-text conflict or a shared harmful target.
Step 2: Adjudicate conflicting evidence, such as a rhetorical tool detecting irony while a knowledge tool reports factual uncertainty.
Step 3: Generate a weighted summary that emphasizes high-confidence and semantically relevant dimensions while preserving unresolved uncertainty.
Output JSON:
{ "supporting_harmful_evidence": [...],
 "supporting_harmless_evidence": [...],
 "conflicts": [...], "unresolved": [...],
 "consensus_summary": "...",
 "evidence_weights": [...] }

CBDF decision prompt

Based on the evidence summary and retrieved label distribution, decide whether the meme is harmful.
Use the following rules:
- If the consensus summary explicitly mentions harmful meme features, such as group-targeted degradation, dehumanizing metaphor, cultural-symbol appropriation, misogynistic implication, or event-grounded harassment, and the reliable retrieved neighbors mostly support harmfulness, treat this as a high-confidence harmful signal.
- If the evidence summary shows benign humor, weak retrieval, no identifiable target, or no cross-modal harmful implication, treat this as harmless or unresolved rather than forcing a harmful label.
- If tool confidence differs by more than 0.30, retrieval reliability is low, or core semantic layers remain unresolved, set human_review to true.
Output JSON:
{ "prediction": "harmful" or "harmless",
 "confidence": 0.xx, "reasoning": "...",
 "evidence_chain": [...], "human_review": true or false }