All Changes May Have Invariant Principles: Improving Ever-Shifting Harmful Meme Detection via Design Concept Reproduction

Anonymous ACL submission

Abstract

Harmful memes are ever-shifting in the Internet communities, which are difficult to analyze due to their type-shifting and temporal-evolving nature. Although these memes are shifting, we find that different memes may share invariant principles, i.e., the underlying design concept of malicious users, which can help us analyze why these memes are harmful. In this paper, we propose REPMD, an ever-shifting harmful meme detection method based on the design concept reproduction. We first refer to the attack tree to define the Design Concept Graph (DCG), which describes steps that people may take to design a harmful meme. Then, we derive the DCG from historical memes with design step reproduction and graph pruning. Finally, we use DCG to guide the Multimodal Large Language Model (MLLM) to detect harmful memes. The evaluation results show that REPMD achieves the highest accuracy with 81.1% and has slight accuracy decreases when generalized to type-shifting and temporal-evolving memes. Human evaluation shows that REPMD can improve the efficiency of human discovery on harmful memes, with 15∼30 seconds per meme.

Disclaimer: This paper may contain content that is disturbing to some readers.

1 Introduction

Nowadays, memes have emerged as important cultural symbols on the Internet. They use both visual and textual elements to convey the designer's opinion for organizations, regimes, and social events (Lin et al., 2024). However, malicious users may design harmful memes as a weapon to express their biased viewpoints (Sharma et al., 2022), which may pose threats to society and other users.

The rapid growth of the Internet community has led to the following ever-shifting characteristics of harmful memes (Valensise et al., 2021): **(1) Type Shifting:** Memes come in many forms and attack different targets, and **(2) Temporal Evolving:** Memes are temporally related to some events. This ever-shifting nature makes the visual elements and expression of new memes quite different from the historical ones. For example, Figure 1's target meme uses a new and implicit way to express discrimination against black people, through the highlighted red circles on the people's accessories (i.e., nose stud). Existing harmful meme detection methods (Kiela et al., 2020; Suryawanshi et al., 2020) only learn the combination of harmful elements, and lack understanding of these implicit expressions, so they cannot accurately detect the harmfulness. Moreover, new slang abbreviations like "*GOAT*" and "*Stan*" are used in memes to express harmful viewpoints, and these rarely-seen expression improve the difficulty of detection.

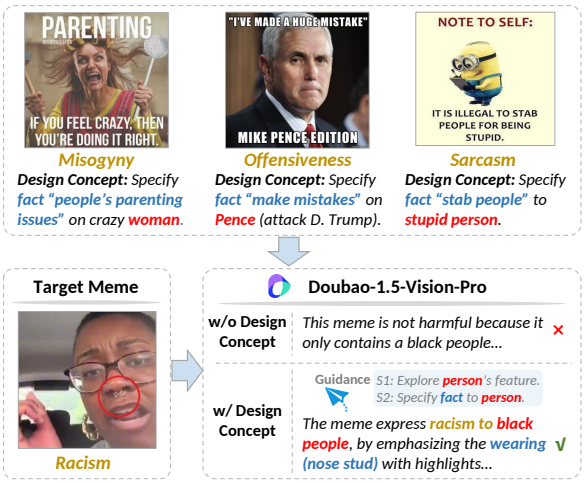

Figure 1: The motivation example of REPMD.

Although harmful memes are ever-shifting, we find that they have **"invariant principles"**. In Figure 1, we choose four example memes that have different harmful types, but share similar underlying ideas that MLLM can learn from each other. For example, the idea of the leftmost meme is to specialize in fact (e.g., parenting issues) to a certain group of people (e.g., crazy women), thereby attacking these people. This is the **Design Concept**

of memes, expressing the attacks through the *human specification* (Sălcudean, 2020). By analyzing the design concepts, the test MLLM, i.e., Doubao-1.5-Vision-Pro (Guo et al., 2025), can accurately identify the target meme's racism: the malicious user specifies a fact "*people who wear nose stud*" onto "*black people*" to express the stereotype.

In the previous cases, we can see that the design concept is useful. However, previous works have not defined an explainable structure to formally describe this design concept. Therefore, we need to define its structure and derive the contents based on the visible information of memes. Moreover, we also need to propose the corresponding usage strategy of design concepts, which can effectively guide the MLLM to detect target harmful memes.

In this paper, we propose an automated approach that **Rep**roduce memes' design concept to improve the ever-shifting harmful **M**eme **D**etection, named **REPMD**. Inspired by the effectiveness of the attack tree (Schneier, 1999), we define the structure of DCG, a **heterogeneous graph** (Zhang et al., 2019; Gao et al., 2025) that describes steps that people may take to design a harmful meme, as well as the goal they want to achieve. This structure can explain the user's design logic and guide MLLM to stepwise identify the harmfulness. Based on this structure, we derived a DCG from historical memes that MLLMs fail to predict accurately, through the design steps' reproduction and graph pruning. For target memes, we retrieve similar reproduction steps from DCG, then form the stepwise guidance to help MLLM detect target harmful memes in the ever-shifting scenario.

To evaluate the performance of REPMD, we conduct two types of experiments on a dataset of 58,192 memes. For the type-shifting experiment on public GOAT-Bench memes, REPMD achieves the highest accuracy with 81.1% and has only a 2.1% accuracy decrease in out-of-domain evaluation. For the temporal-evolving experiment on our manually crawled Twitter memes, REPMD also outperforms baselines and has a 0.3% accuracy improvement in other quarters' evaluation. Moreover, our human evaluation shows that REPMD can improve the efficiency of human discovery on harmful memes, with 15∼30 seconds per meme.

This paper makes the following contributions:

- We propose REPMD, a harmful meme detection method with design concept reproduction, applicable for ever-shifting memes on the Internet.
- We evaluate REPMD on ever-shifting memes from two data sources, which outperforms baselines and can be generalized to type-shifting and temporal-evolving memes with the help of DCG.
- We conduct a human evaluation to illustrate that DCG has high explainability and can help evaluators manually identify harmful memes.
- We release the code and dataset[1] to facilitate REPMD's reproducibility.

2 Definition of Design Concept

To describe the meme's design concept, we refer to the idea of attack trees (Schneier, 1999), a threat model that contains STRIDE threat types, attack methods, goals, and logic of the attack reproduction. It can guide LLM-based security testing tools to warn of software vulnerabilities (Xiong and Lagerström, 2019), which aligns with MLLM's logic for identifying memes' harmful information.

Before extracting the DCG, we first analyze the reasons why harmful memes cannot be detected by MLLM, then form the fail reason tree. It contains the type and fail reason nodes, as well as the type and link edges. Then, we derive the fail reason node to the steps that the harmful users will take to design harmful memes, which contain the type, reproduction method, goal, and logic gate nodes, as well as the type, link, and achievement edges.

2.1 Definition of Fail Reason Tree

The structure of fail reason tree can be formed as $\mathcal{G}_{\mathrm{F}} = \langle \{\mathcal{N}_{\mathrm{T}}, \mathcal{N}_{\mathrm{F}}\}, \{\mathcal{E}_{\mathrm{T}}, \mathcal{E}_{\mathrm{Link}}\} \rangle$, defined as follows: **Node: (1) Type Node $\mathcal{N}_{\mathbf{T}}$:** It contains three node levels $L_1 \sim L_3$, where the chosen value is in Table 5, similar to STRIDE threat types in attack trees: **Macro Type Nodes $\mathcal{N}_{\mathbf{T}}^{L_1}$:** The seven fixed categories in Table 5, defined in the previous works (Chen et al., 2025); **Subtype Nodes $\mathcal{N}_{\mathbf{T}}^{L_{2,3}}$:** The detailed types that need to be specified, e.g., *Culture→Video Games*. These subtypes are **not limited to the table** and can be extended. **(2) Fail Reason Node $\mathcal{N}_{\mathbf{F}}$:** It contains one-sentence MLLM's fail reason and memes' text description. **Edge: (1) Type Edge $\mathcal{E}_{\mathbf{T}}$:** The edge is $\mathcal{N}_{\mathrm{T}}^{L_i} \rightarrow \mathcal{N}^{L_{i+1}}$, indicating the division from the macro type to subtypes; **(2) Link Edge $\mathcal{E}_{\mathbf{Link}} = \mathcal{N}_{\mathbf{T}} \rightarrow \mathcal{N}_{\mathbf{F}}$:** It indicates that one type of incorrectly predicted harmful meme has corresponding fail reasons.

[1] https://anonymous.4open.science/r/RepMD-448C

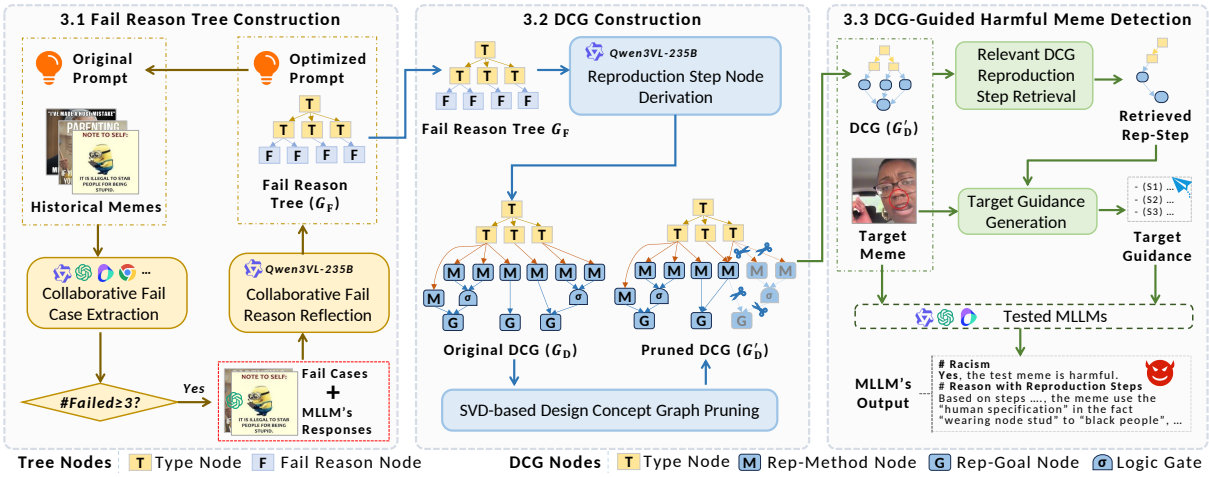

Figure 2: The overview of REPMD.

2.2 Definition of DCG

We extract design concepts of historical memes and gather them into the DCG. The structure refers to the attack tree's three-level structure, i.e., attack method, attack goal, and logic gate, which reflect how to reproduce a harmful meme. The structure of DCG can be formed as $\mathcal{G}_{\mathrm{D}} = \langle\{\mathcal{N}_{\mathrm{T}}, \mathcal{N}_{\mathrm{M}}, \mathcal{N}_{\mathrm{G}}, \mathcal{N}_{\sigma}\}, \{\mathcal{E}_{\mathrm{T}}, \mathcal{E}_{\mathrm{A}}, \mathcal{E}_{\mathrm{Link}}\}\rangle$, where the reproduction nodes are derived from reason nodes $\mathcal{N}_{\mathrm{F}} \rightarrow (\mathcal{N}_{\mathrm{M}}, \mathcal{N}_{\mathrm{G}}, \mathcal{N}_{\sigma})$, which is the reproduction step for the DCG. defined as follows:

Node: (1) Type Node $\mathcal{N}_{\mathrm{T}}$: They are the same as the fail meme tree. **(2) Reproduction Method $\mathcal{N}_{\mathrm{M}}$:** These nodes describe the steps that malicious users take to design harmful memes; **(3) Logic Gate $\mathcal{N}_{\sigma}$:** There is a logical combination between the reproduction methods to design a meme, i.e., *And* ($\wedge$), *Or* ($\vee$), and *Not* ($\neg$); **(4) Reproduction Goal $\mathcal{N}_{\mathrm{G}}$:** A design goal that a malicious user aims to achieve, such as the *human specification*. Each node also contains a harmful indicator $\{0, 1\}$ that identifies which node may pose a harmful.

Edge: (1) Type Edge $\mathcal{E}_{\mathrm{T}}$: It is the same as the fail meme tree. **(2) Link Edge $\mathcal{E}_{\mathrm{Link}} = \mathcal{N}_{\mathrm{T}} \rightarrow \mathcal{N}_{\mathrm{M}}$:** The structure is the same as the fail meme tree, but the meaning is slightly different. The link indicates that in historical memes, one type of harmful meme has the corresponding reproduction step. **(3) Achievement Edge $\mathcal{E}_{\mathrm{A}}$:** Edges between $(\mathcal{N}_{\mathrm{M}}, \mathcal{N}_{\mathrm{G}}, \mathcal{N}_{\sigma})$, which are the logic indicating the achievement between methods and the goal.

3 Overview of REPMD

The element changing in ever-shifting harmful memes makes them difficult to detect. As shown in Figure 1, the target meme expresses racism against people in an unknown and implicit way. With the help of the design concepts, we find that different types of memes may have invariant principles, such as the above-mentioned *human specification*. This invariant feature can help MLLMs infer why the unknown meme is harmful.

Figure 2 illustrates the overview of REPMD. Guided by the concept, we first explore the reasons in historical meme cases that MLLMs fail to detect harmful memes, formed as the fail reason tree defined in Section 2. Second, we derive reproduction steps from the fail reason tree and form DCG, then propose a fast **Singular Value Decomposition (SVD)**-based graph pruning method, proved effective in GNN's dimensionality reduction (Cai et al., 2023). Finally, REPMD retrieves relevant reproduction steps from DCG to help MLLMs detect whether the target meme is harmful.

3.1 Fail Reason Tree Construction

Before constructing the DCG $\mathcal{G}_{\mathrm{D}}$, we aim to explore the reason why MLLMs fail to detect harmful memes. Given historical memes $[M_1^H, ..., M_n^H]$, we propose an looped framework of "*extraction, reflection, and optimization*". After the loop, we can obtain a fail reason tree $\mathcal{G}_{\mathrm{F}}$ and the optimized prompt. Through prompt optimization, the fail reason tree will contain some harmful memes that cannot be accurately detected, no matter how the prompt is optimized. Therefore, it is meaningful to explore design concept for these memes.

3.1.1 Collaborative Fail Case Extraction

For each historical meme M_i^H, given the original P_{Harm} (see Figure 9), we input it into the five tested MLLMs, i.e., Doubao-1.5-Vision, GPT-4o, Qwen2.5VL, Gemini-Pro, and InternVL3, based on evaluation reports (Lu et al., 2024; Li et al., 2024a). These MLLMs achieve state-of-the-art

performances in tasks like OCR and Image Understanding, etc., with high **Performance** and **Time Efficiency**. Some novel models, like Qwen3VL-235B, may take over two minutes with a long chain-of-thought, so we do not evaluate them. We then obtain fail cases M_{fail} with the major voting, i.e., ≥ 3 MLLMs Fail, which means in the i_{th} iteration, even prompts are optimized, MLLMs still struggle to identify their harmfulness, so they need to be prioritized to analyze.

3.1.2 Collaborative Fail Reason Reflection

Given fail cases M_{fail}, we use the tree generation MLLM with prompt P_{F} (see Figure 10) to generate $\mathcal{G}_{\text{F}}$: **(1) Reason Nodes Generation:** We concatenate the responses of all fail MLLMs, then ask a larger MLLM (i.e., Qwen3VL-235B) to analyze why they cannot detect the harmful: $M_{\text{fail}} \rightarrow \mathcal{N}_{\text{F}}$. We directly tell Qwen3VL that MLLMs fail in prompt P_{F} and ask it to generate content in the fail reason nodes. **(2) Type Nodes Generation:** To analyze which type the meme belongs to, we input the pre-defined types and the reason node's content into Qwen3VL and categorize it into type nodes: $(M_{\text{fail}}, \mathcal{N}_{\text{F}}) \rightarrow \mathcal{N}_{\text{T}}$. The prompt controls MLLM to first classify the meme into the macro type as L_1, then stratify subtypes from L_2 to L_3. **(3) Prompt Optimization.** We use Qwen3VL to summarize the key points in $\mathcal{G}_{\text{F}}$, which summarize the common causes of fail memes and what the MLLM needs to focus on. We optimize the prompt by appending these points: $P_{\text{Harm}} \oplus \text{KeyPoints} \rightarrow P'_{\text{Harm}}$, which makes MLLMs focus on fail cases and pertinently improve the performance of meme detection.

3.2 DCG Construction

Given the fail reason tree $\mathcal{G}_{\text{F}}$ with the "reason-level expression" (e.g., Figure 5's "*crazy woman with the parenting issues*"), we aim to explore memes' "reproduction-level idea" (e.g., "*deliberately choose a crazy woman to emphasize and attacking these persons*). The "reproduction" means thinking about the user's design process based on visible information. Its core idea is to find a harmless alternative element, thereby identifying the reason why users choose the current element.

We propose the node derivation and calibration method to derive the reproduction steps and form the original DCG: $\mathcal{G}_{\text{D}}$. Then, since $\mathcal{G}_{\text{D}}$ may contain redundant nodes, and MLLMs may have excessive time cost to prune this graph, we propose an effective pruning algorithm based on the SVD.

Algorithm 1: SVD-based DCG pruning.

Input: The original DCG $\mathcal{G}_{\text{D}}$, retained proportion θ.
Output: The pruned DCG $\mathcal{G}'_{\text{D}}$.

1 $\mathbf{A} = Score_{\text{rep}}(\mathcal{G}_{\text{D}})$ with Equation (1), where $\mathbf{A} = \begin{pmatrix} \mathbf{A}_{\text{Type}} & \beta\mathbf{E}_{\text{Link}} \\ \beta\mathbf{E}_{\text{Link}} & \alpha\mathbf{A}_{\text{Repr}} \end{pmatrix}$, α, β: scaled factors;
2 Calculate Degree Matrix $\mathbf{D}$, Initialize $\mathcal{G}'_{\text{D}} = \emptyset$;
3 **while** $t \leq 5$ **do**
4 $\quad \mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}^t\mathbf{D}^{-1/2}$;
5 $\quad$ SVD: $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top$ (See Appendix A.2.1); $\Lambda = \text{diag}(\lambda_1, ..., \lambda_n), \lambda_1 \geq \lambda_2 \geq ... \geq 0$;
6 $\quad$ Cut-off Determination (See Appendix A.2.2): $cut = \arg\max_i |\ln(\lambda_{i+1} - \lambda_i)|$;
7 $\quad$ Graph Pruning: $\mathcal{G}'_{\text{D}} = \langle \{\mathcal{N}_1, ..., \mathcal{N}_{cut}\}, \{\mathcal{E}'\} \rangle$, where $\{\mathcal{N}_1, ..., \mathcal{N}_{cut}\} \notin \mathcal{G}'_{\text{D}}$;
8 $\quad$ **if** $cut/n \geq \theta$ **then**
9 $\quad\quad$ **break**;
10 $\quad$ **end**
11 $\quad$ Analyze Indirect Features: $t = t + 1$;
12 **end**
13 return $\mathcal{G}'_{\text{D}}$;

3.2.1 Reproduction Step Node Derivation

For reason nodes $\mathcal{N}_{\text{F}}$, we derive reproduction steps with Qwen3VL-235B and prompt P_{D} (see Figure 11): $\mathcal{N}_{\text{F}} \rightarrow \mathcal{G}_{\text{D}}$: **(1) Reproduction Method and Logic Gate Derivation:** We ask Qwen3VL to capture the reason-level elements with the combination logic $\mathcal{N}_\sigma$. Then, according to the previous description of alternative design approaches, we drill down into the ideas and steps that focus on the method of designing this meme $\mathcal{N}_{\text{M}}$, by asking Qwen3VL three questions: "***A:** Is there a replacement method for each element?*", "***B:** Why is that element chosen?*", and "***C:** Is the replaced element harmful?*". **(2) Graph Calibration and Goal Derivation:** We ask another Qwen3VL to validate whether the design concept aligns with the meme's visual elements and analyze what the user wants to achieve as $\mathcal{N}_{\text{G}}$. After generating design concepts from historical memes, we link them by type nodes and form an overall graph as DCG.

3.2.2 SVD-based DCG Pruning

The DCG is the gathering of all historical memes, so some memes may have similar design concepts, introducing redundant nodes and edges to the DCG. Therefore, we propose the SVD-based pruning method, as shown in Algorithm 1. The main idea is how to design an **adjacency matrix** to describe the relations between nodes $\mathcal{N}_i$ and $\mathcal{N}_j$, so we propose the reproduction score (**line 2**):

$$Score_{\text{rep}}(\mathcal{N}_{ij}) = \text{ReLU}(sim(\mathcal{N}_i, \mathcal{N}_{root}) - sim(\mathcal{N}_i, \mathcal{N}_j)) \quad \text{where } sim(\mathcal{N}_i, \mathcal{N}_j) = 0 \text{ if } \mathcal{N}_i/\mathcal{N}_j = \mathcal{N}_{\text{root}} \tag{1}$$

where the function $sim(\cdot,\cdot)$ is the cosine similarity between TF-IDF values of the node's contents. $\mathcal{N}_{\text{T}}$'s root node is $\mathcal{N}_{\text{T}}^{L_1}$, and $\mathcal{N}_{\text{M}}$'s root node is $\mathcal{N}_G$. The final scores are activated with the ReLU function (Glorot et al., 2011) to ensure the positive definiteness of the adjacent matrix. The equation implies that we want a node to have **higher correlation** to the root node and **lower similarity** to other nodes, which means that this node is not redundant and will not be pruned. Moreover, we find that redundancy mainly comes from reproduction steps, so we introduce scaled factors α, β for the matrix of the reproduction step $(\mathcal{N}_{\text{M}}, \mathcal{N}_{\text{G}}, \mathcal{N}_{\sigma}) \rightarrow \mathbf{A}_{\text{Repr}}$ and the edge matrix $\mathcal{E}_{\text{Link}} \rightarrow \mathbf{A}_{\text{Link}}$ $(1 > \beta > \alpha)$.

From **line 3** to **line 12**, we set a hyperparameter θ to represent the retained nodes in $\mathcal{G}_{\text{D}}^{'}$. We first introduce the Laplacian Normalization (Li et al., 2024b) to process the adjacent matrix to $\mathbf{L}$. Then, we conduct the SVD and obtain singular values $[\lambda_1, ...\lambda_n]$, which can be used to represent the matrix's feature with the low-rank approximation (**line 5**). Since the distribution of singular values may be long-tailed, we calculate the difference on a logarithmic scale and choose the cut-off value λ_{cut} with the steepest decline. Finally, if the proportion of retained nodes does not achieve θ, we explore the impact of indirect relations (i.e., $\mathbf{A}^t$).

3.3 DCG-Guided Harmful Meme Detection

Since REPMD has constructed DCG with multiple prompts, the detection is lightweight by reusing previous prompts. Given target memes $[M_1^T, ..., M_n^T]$ and the test MLLM, we retrieve the similar reproduction steps from the DCG $\mathcal{G}_R^{'}$ and guide the harmful meme detection. Since this step is executed online, the time cost of REPMD comes from the scale of the test dataset.

Step-1: Relevant DCG Reproduction Step Retrieval. We extract the historical reproduction step as follows: **(1) DCG Node Retrieval:** We reuse the prompt P_{F} and P_{D} to select DCG's nodes that may relate to M_i^T, aggregated as a set $\{\mathcal{N}_i | \mathcal{N}_i \in \mathcal{G}_{\text{D}}^{'}\}$. **(2) Reproduction Step Formation:** We cluster the subgraphs based on the edges, i.e., $\mathcal{G}_{\text{sub}} = \langle \{\mathcal{N}_i\}, \mathcal{E}_{\text{sub}} \subset \mathcal{G}_{\text{D}}^{'} \rangle$.

Step-2: Target Guidance Generation. We transform the retrieved step $\mathcal{G}_{\text{sub}}$ into plain text with logic semantics (see Appendix A.1), then feed the target meme and this plain text into the test MLLM (same model for testing the follow-up harmful meme detection) to output the guidance as $(S_1) \rightarrow (S_n)$. Finally, REPMD inputs the generated guidance with the target meme M_i^T into the test MLLM, then uses the optimized prompt $P_{\text{Harm}}^{'}$ in Figure 9 to detect the target harmful memes.

Table 1: The statistics of our constructed dataset.

Types	**Type-Shifting Dataset from GOAT-Bench**				
	Racism	Misogyny	Offensiveness	Sarcasm	Toxity
DCG	8,310	10,190	7,000	19,816	4,250
Target	2,080	920	743	1,820	1,063
Quarters	**Temporal-Evolving Dataset from Twitter**				***Total***
	Jan∼Apr	Apr∼Jun	Jul∼Sep	Oct∼Dec	
DCG	400	400	400	400	*51,166*
Target	100	100	100	100	*7,026*

4 Experimental Design

To evaluate the performance of REPMD, we introduce three Research Questions (RQs).

RQ1: *What are the performances of* REPMD *on detecting ever-shifting harmful memes, i.e., type-shifting or temporal-evolving memes.*

RQ2: *How do each component contribute to* REPMD*'s performances?*

RQ3: *Can* REPMD*'s DCG help humans understand why the memes are harmful?*

Dataset Preparation. We construct the dataset from two sources, i.e., GOAT-Bench (Lin et al., 2024) and our crawled memes from Twitter, as follows. **(1) Type-Shifting Dataset Preprocessing:** We first reuse all memes in GOAT-Bench. Then, we manually find that it has overlap categories (misogyny and hatefulness) and misclassified samples (classify misogyny memes as hatefulness), so we replace the type name, e.g., "Harmfulness"→"Toxicity" (violence and suicide, etc) and "Hatefulness"→"Racism" (mainly the racial hatefulness). Then, we reclassify 110 misogyny memes to the correct type. The revised data is used to detect type-shifting harmful memes. **(2) Temporal-Evolving Dataset Annotation:** There are currently no usable benchmarks, so we manually crawl the memes on Twitter in **year: 2025** and divide them into four quarters. In each quarter, we randomly select 500 memes for DCG construction and evaluation. We have invited three researchers with ≥ 3 years of experience in harmful meme detection (not included in the authors) to annotate the dataset independently and check whether the labels are accurate. The average Cohen's Kappa (Pérez et al., 2020) value is 0.86, which achieves a high agreement on the labels. Table 1 shows that we collect 58,192 memes (54.0% are harmful) in the dataset, where 56,192 comes from GOAT-Bench

Table 2: The performance of REPMD on In-Domain (ID) and Out-Of-Domain (OOD) memes(%).

Evaluation on Types	**Racism**		**Misogyny**		**Offensiveness**		**Sarcasm**		**Toxicity**		***Average***	
Models	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.
Few-Shot SFT and RAG-based Baselines for Harmful Meme Detection												
Mod-Hate$_{ID}$ (Cao et al., 2024)	58.3	58.1	61.9	61.5	63.0	64.1	61.3	63.3	71.2	69.3	63.1	63.3
Mod-Hate$_{OOD}$	36.6	34.9	43.7	44.5	40.2	43.7	42.1	45.2	59.3	55.1	44.4	44.7
MIND$_{ID}$ (Liu et al., 2025)	69.1	66.2	69.2	68.7	63.7	62.5	64.0	63.2	71.6	71.2	67.5	66.4
MIND$_{OOD}$	47.4	45.0	49.5	49.4	42.1	42.0	45.9	44.3	55.3	54.0	48.0	46.9
RA-HMD$_{ID}$ (Mei et al., 2025)	74.6	73.4	77.3	76.8	71.9	70.1	70.4	71.3	70.2	69.5	72.9	72.2
RA-HMD$_{OOD}$	57.2	56.3	50.2	48.6	48.0	47.4	45.2	44.3	64.5	62.3	55.0	51.8
Average Δ(OOD, ID)	↓20.3	↓20.5	↓21.7	↓21.5	↓22.8	↓21.2	↓20.8	↓21.3	↓11.3	↓12.9	↓18.7	↓19.5
Test MLLMs for Harmful Meme Detection												
Qwen2.5VL-32B (Vanilla*)	72.6	68.3	65.2	60.9	63.2	55.3	71.0	68.7	62.3	53.9	66.9	61.4
+REPMD$_{ID}$	78.2	77.4	74.9	74.5	71.0	70.6	80.4	79.6	79.1	77.0	76.7	75.8
+REPMD$_{OOD}$	76.8	75.4	73.0	72.6	70.2	70.1	77.2	77.2	77.5	75.8	74.9	74.2
GPT-4o (Vanilla)	76.6	75.2	82.7	82.4	59.1	59.1	68.7	66.3	66.3	64.2	70.7	69.4
+REPMD$_{ID}$	**88.5**	**87.6**	**90.5**	**90.4**	71.2	71.1	**84.2**	**83.8**	80.2	78.5	**82.9**	**82.3**
+REPMD$_{OOD}$	87.1	87.0	88.7	86.2	70.4	70.2	80.5	79.6	77.6	76.9	80.9	80.0
Doubao-1.5-Vision-Lite (Vanilla)	70.5	69.1	67.3	68.0	66.4	65.5	68.6	68.7	70.0	70.2	68.6	68.3
+REPMD$_{ID}$	77.4	77.2	74.5	74.1	76.3	75.4	75.0	74.9	78.6	78.5	76.4	76.0
+REPMD$_{OOD}$	76.2	75.5	73.2	73.7	72.1	73.1	74.1	72.6	77.5	76.2	74.6	74.2
Doubao-1.5-Vision-Pro (Vanilla)	72.4	71.9	73.4	72.9	67.4	64.3	71.3	70.4	72.0	72.1	71.1	69.9
+REPMD$_{ID}$	81.5	80.6	82.2	81.3	**80.2**	**81.3**	81.2	81.3	**83.0**	**82.8**	81.3	81.1
+REPMD$_{OOD}$	80.3	79.4	80.4	79.3	78.2	77.6	78.5	78.4	81.7	80.9	79.4	78.7
Average Δ(ID, Vanilla)	↑7.7	↑9.0	↑7.6	↑7.9	↑9.4	↑12.8	↑9.2	↑9.8	↑11.7	↑13.1	↑9.1	↑10.5
Average Δ(OOD, ID)	↓1.3	↓1.4	↓1.7	↓2.1	↓2.0	↓1.9	↓2.6	↓3.0	↓1.7	↓1.7	↓1.9	↓2.1

* "Vanilla" means using original MLLMs with the optimized detection prompt $P_D^{'}$ to detect harmful memes.

for type-shifting evaluation, and 2,000 are crawled memes for temporal-evolving experiments.

Baselines. We select three novel and representative baselines. One method utilizes Retrieval-Augmented Generation (RAG) for harmful meme detection, which meets the evaluation requirements of DCG construction and retrieval. The other two methods introduce few-shot Supervised Fine-Tuning (SFT), which meet the experimental requirements on ever-shifting memes: **Mode-Hate** (Cao et al., 2024) is a few-shot hateful meme detector fine-tuned with LoRA; **MIND** (Liu et al., 2025) is a zero-shot RAG-based framework with bidirectional insight-augmented inference; and **RA-HMD** (Mei et al., 2025) is a fine-tuned adaptation framework for hateful meme detection. Moreover, we have also selected four representative MLLMs as baselines, i.e., **Qwen2.5VL-32B** (Bai et al., 2025), **GPT-4o**, **Doubao-1.5-Vision-Lite**, and **Doubao-1.5-Vision-Pro** (Guo et al., 2025).

Metrics. To ensure a fair and comprehensive evaluation, we measure model performance using two commonly-used metrics: **Accuracy** and **F1-Score**, following the setting of GOAT-Bench.

Experimental Settings. For implementation, we use MLLM's APIs and set $temperature = 0$ to make the output fixed, Top_P, and max_token as default values in the vanilla MLLM. For hyperparameters, we set $\alpha = 0.3$ and $\beta = 0.6$ for SVD, and $\theta = 75\%$ (see Appendix A.4). All experiments run on GeForce RTX A6000 GPU.

5 Results

5.1 Overall Performance of REPMD on Evolving Memes (RQ1)

Type-Shifting Experiments. We conduct this experiment in the following types: **(1) In-Domain (ID) Evaluation:** We construct DCG and detect target memes' harmfulness within the same type. **(2) Out-Of-Domain (OOD) Evaluation:** We introduce the cross-type analysis, which means that we choose a specific type of memes in GOAT-Bench as the target, and use **the other four types** to generate DCG. Table 2 illustrates the results for type-shifting scenario, where Δ(ID, Vanilla) and Δ(OOD, ID) indicate the absolute improvement of REPMD and the relative decrease when migrated to the OOD memes. We can see that, the absolute detection performance of REPMD achieves 81.1% accuracy, outperforming all baseline approaches. The improvement on vanilla MLLMs achieves +9.1% (F1) and +10.5% (Accuracy). Besides, the OOD experiment shows that the decrease of REPMD's performance is smaller than baselines (which is -18.7% F1), with -1.9% (F1) and -2.1% (Accuracy). By analyzing the inference of MLLM, we find that they detect harmful memes by analyzing what elements

Table 3: The performance of REPMD on Temporal Evolving (TE) and Temporal Fixed (TF) memes (%).

Evaluation on Quarters Models (Top-2 MLLMs)	**Apr∼Jun*** F1	Acc.	**Jul∼Sep** F1	Acc.	**Oct∼Dec** F1	Acc.
RA-HMD$_{TF}$	45.9	44.9	52.7	51.5	56.4	55.2
RA-HMD$_{TE}$	31.7	31.2	40.2	40.0	45.3	44.1
Δ(TE, TF)	↓14.2	↓13.7	↓12.5	↓11.5	↓11.1	↓11.1
GPT-4o (Vanilla)	51.3	50.0	56.3	55.0	67.4	65.0
+REPMD$_{TF}$	60.2	60.0	63.0	66.0	83.5	80.0
+REPMD$_{TE}$	67.4	65.0	61.6	65.0	82.4	78.0
Δ(TF, Vanilla)	↑8.9	↑10.0	↑6.7	↑11.0	↑16.1	↑15.0
Δ(TE, TF)	↑7.2	↑5.0	↓1.4	↓1.0	↓1.1	↓2.0
Doubao-1.5-V-Pro (Vanilla)	67.4	65.0	66.2	65.0	67.4	65.0
+REPMD$_{TF}$	82.4	80.0	82.4	80.0	86.7	85.0
+REPMD$_{TE}$	82.4	80.0	82.4	80.0	86.7	85.0
Δ(TF, Vanilla)	↑15.0	↑15.0	↑16.2	↑15.0	↑19.3	↑20.0
Δ(TE, TF)	0.0	0.0	0.0	0.0	0.0	0.0

* The ***year: 2025*** is divided into quarter-based intervals.

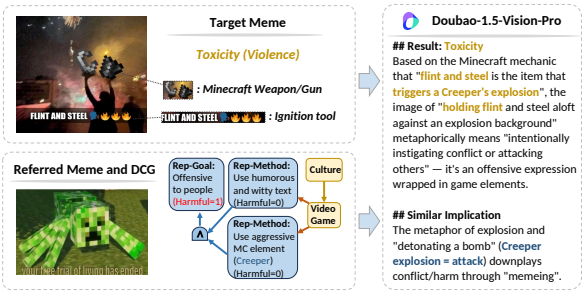

Figure 3: The case study of temporal evolving memes.

are harmful, but may ignore the implicit expressions (e.g., Figure 1's red circle) and seemingly harmless elements. From the experiment results, we can see that DCG guides MLLM to understand the idea of memes' design, enabling REPMD to detect harmful memes across types. Even if there are not too many training samples labeled with fine-grained types, REPMD can achieve effective detection ability by $\{0, 1\}$ harmful labels.

Temporal-Evolving Experiments. We conduct this experiment in the following types: **(1) Temporal Fixed (TF) Evaluation:** We construct DCG and evaluate REPMD on same quarter's memes. **(2) Temporal Evolving (TE) Evaluation:** To ensure fairness of evaluation, we choose **quarter**$_i$'s cases to construct DCG and detect harmful memes in **quarter**$_{i+1}$'s cases. Table 3 shows the Top-2 MLLMs' performances on Twitter's temporal-evolving memes within the last three quarters. We can see that, REPMD improves the vanilla MLLMs with +13.7% (F1) and +14.3% (Accuracy) on average. Moreover, we calculate the difference Δ(TE, TF) and find that introducing historical memes can even improve the detection performance of memes, with +0.8% (F1) and +0.3% (Accuracy) on average. This advantage comes from the consistency with the design concepts of the same organization. Some subcultural groups (e.g., Bitcoin's Dogwifhat, Minecraft, and K-pop Fancam, etc.) have opinion leaders, and group members will utilize a similar expression pattern to design memes, so the continuity of these patterns can help us identify the causes of harmful memes accurately. Figure 3 shows that the Minecraft group's users use elements like "*Creeper*" and "*Ignition Tool*" instead of regular weapons to express violence, This expression is successfully captured by REPMD in the DCG, so it can help the MLLMs understand ideas of the group's opinion leaders and members, thus detecting harmful memes that are rapidly shifting.

Table 4: The accuracy of REPMD's ablation study (component removal or replacement) on three variants (%).

Variants	**ID**	**OOD**	**TE**
GPT-4o+REPMD	82.3	80.0	69.3
-VoteMLLMs	81.0 (↓1.3)	79.2 (↓0.8)	65.3 (↓4.0)
-OptPrompt	79.0 (↓3.3)	78.6 (↓1.4)	64.9 (↓4.4)
DCG $\mathcal{G}_D$ →Tree $\mathcal{G}_F$	76.8 (↓5.5)	71.6 (↓8.4)	60.5 (↓8.8)
SVD→GPT-4o	79.9 (↓2.4)	78.2 (↓1.8)	68.1 (↓1.2)
SVD→Doubao-V-Pro	*Failed*	79.1 (↓0.9)	69.0 (↓0.3)
SVD→Qwen3VL	*Failed*	*Failed*	70.0 (↑0.7)
-Retrieval	76.3 (↓6.0)	70.9 (↓9.1)	60.1 (↓9.2)
Retrieval→ImgSim	78.2 (↓4.1)	77.1 (↓2.9)	64.5 (↓4.8)
Retrieval→GraphRAG	81.1 (↓1.2)	79.5 (↓0.5)	65.5 (↓3.8)

5.2 Ablation Study of REPMD (RQ2)

To evaluate the contribution of model's components to REPMD, we analyze three types of **variants** based on the structure in Section 3: **Sec 3.1:** Removing the vote mechanism (**-VoteMLLMs**) and prompt optimization (**-OptPrompt**); **Sec 3.2:** Replacing DCG with fail reason tree (**DCG→Tree**) and SVD with MLLMs (**SVD→MLLMs**); **Sec 3.3:** Directly generating reproduction steps without retrieval (**-Retrieval**), replacing retrieval with image similarity (**Retrieval→ImgSim**), and the most similar graph-based RAG method, i.e., (**Retrieval→GraphRAG** (Peng et al., 2024)).

Table 4 shows the results of the ablation study. We can see that, REPMD outperforms most variants (5/6) in terms of the experiments ID, OOD, and TE. The maximum decrease is in the components that are related to the DCG, i.e., *DCG→Tree* and *-Retrieval*, which achieve over -8.0% and -9.0% Accuracy. Considering *SVD→MLLM*, we find that replacing SVD with Qwen3VL-235B fails on ID and OOD because of information truncation (too many input tokens) and incorrect output (pruned graph's nodes are different from original DCG). However, MLLM successfully prunes the graph and slightly outperforms the SVD in the TE task, with over 14× time cost (see Appendix A.4). This indicates that SVD has advantages when balancing

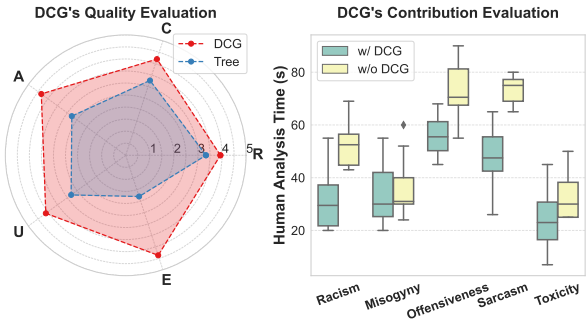

Figure 4: The results of human evaluation.

time cost and performance, but if we can fine-tune an LLM-based pruning method, it may improve DCG's quality and detection performance.

5.3 Human Evaluation of REPMD (RQ3)

We conduct the human evaluation on the DCG in two aspects: **(1) DCG's Quality:** We have invited three types of human evaluators categorized by research experiences on harmful memes (i.e., $\geq$ 3 year veterans, 1 $\sim$ 3 year beginners, and non-experienced social network users). Each category includes two persons. We randomly select 10 historical memes with their corresponding fail reason trees and the DCG's design concepts, then rate these patterns on a 5-point Likert scale (from 1: poor to 5: excellent) based on five key criteria (Mazhar et al., 2025): *Relevance (R)*, *Correctness (C)*, *Actionability (A)*, *Uniqueness (U)*, and *Explainability (E, which is newly introduced based on DCG's structure)*. **(2) DCG's Contribution:** We randomly select another 10 target harmful memes with generated target guidance, then calculate the time costs that previous evaluators take to identify their harmfulness with/without DCG's help. Note that we only count samples that can be successfully analyzed by humans. We have presented the questionnaire in Appendix A.5.

Figure 4 shows the results of human evaluation. We can see that, for the overall DCG's quality evaluation, the DCG's five-dimensional scores outperform the fail reason trees. All the scores are nearly or over 4, which means the DCG has high uniqueness and can be used to guide the detection of harmful memes. Moreover, the explainability of the DCG (score: 4.4) outperforms that of the trees (score: 1.8) by the largest margin (score: +2.6), which means the DCG is easier to understand by evaluators with different experiences. For the DCG's contribution, the time costs of human evaluation show that DCG can help evaluators improve their efficiency in accurately analyzing why memes are harmful. Especially for the Offensiveness and Sarcasm memes that implicitly express the harmfulness, and humans cannot easily find their harmful elements, the DCG can reduce the average time costs from 15$\sim$30 seconds per meme. Therefore, REPMD can help humans identify harmful memes. Experiments in Appendix A.7 show that DCG can also help safeguard text-to-image models.

6 Related Works

Harmful meme detection has emerged as an important research direction in multimodal semantic analysis. Previous studies extract visual and textual feature embeddings and fuse them with attention mechanisms (Suryawanshi et al., 2020; Pramanick et al., 2021; Lee et al., 2021) and Transformer-based intermediate fusion (Kiela et al., 2019; Lu et al., 2019; Li et al., 2019; Chen et al., 2020), and contrastive reweighting approaches (Kiela et al., 2020; Muennighoff, 2020; Lippe et al., 2020). With the development of MLLMs, researchers proposed prompt-based (Cao et al., 2022; Ji et al., 2023; Cao et al., 2023; Ji et al., 2024) and multi-agent-based frameworks for deeper visual understanding (Lin et al., 2025). Considering the bias in MLLM's inference steps, researchers proposed LLM debate frameworks (Hee et al., 2022) to improve the detection performance. To address challenges in low-resource generalization, researchers have explored few-shot enhanced methods with SFT and RAG methods (Cao et al., 2024; Huang et al., 2024; Liu et al., 2025). However, harmful memes are ever-shifting on the Internet, but few of these works concerns on improving their generalizability. Different from these works, REPMD introduces the explainable design concepts to facilitate the detection of ever-shifting harmful memes.

7 Conclusion

In this paper, we propose REPMD, an ever-shifting harmful meme detection method. We first define DCG, including the harmful types and reproduction steps of how users design harmful memes. Then, we derive the DCG from historical memes. Finally, we retrieve and utilize it to guide the MLLMs to detect whether the target meme is harmful. The evaluation results show that REPMD can detect harmful memes with 81.1% accuracy and can be generalized to type-shifting and temporal-evolving memes. Human evaluation shows that REPMD can improve the efficiency of human discovery on harmful memes, with 15$\sim$30 seconds per meme.

Limitations

Although REPMD is effective in detecting ever-shifting harmful memes, there are some cases that REPMD cannot accurately identify their harmfulness. We manually investigate these bad cases and discuss the reasons for their failures.

First, 60% of the bad cases come from overly simple meme features. For example, the Minecraft group users always use the cubes to represent humans and objects, but MLLMs have a limited ability to learn these expressions. In the future, we plan to combine the MLLM fine-tuning and DCG to learn these features. Second, 35% of the cases come from the LLM hallucinations. MLLM may violate the guidance of DCG during the inference process and output guessed results, leading to errors in harmful meme detection. We plan to introduce the stepwise hallucination corrector to revise the correct biases in the responses. Finally, 5% of the cases come from newly designed memes with completely unseen design concepts. The proportion of these memes is small, which has relatively less impact on the detection performance. We also plan to continuously improve DCG for these cases.

Ethical Statement

Our study aims to detect ever-shifting harmful memes on the Internet, but it may raise ethical considerations. We will discuss how these considerations are mitigated as follows. First, the dataset may have privacy and license issues because it comes from GOAT-Bench and online Twitter posts. We control permissions of visitors to the dataset, through providing private dataset links and password access, and continuously track the whereabouts of the data to prevent data abuse. Second, the algorithms may have fairness issues because there are differences in the protection of individual freedom of speech in different countries. We publicly disclose the DCG and its related guidance, and plan to provide channels for misjudgment on our public website. Third, the annotators and evaluators may have mental health issues when labeling the dataset. We have provided regular psychological counseling to annotators, established a job rotation system, and provided reasonable salaries.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923.

Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. Lightgcl: Simple yet effective graph contrastive learning for recommendation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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*, pages 5244–5252.

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.

Rui Cao, Roy Ka-Wei Lee, and Jing Jiang. 2024. Modularized networks for few-shot hateful meme detection. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 4575–4584, New York, NY, USA. Association for Computing Machinery.

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 *European conference on computer vision*, pages 104–120. Springer.

Zixin Chen, Hongzhan Lin, Kaixin Li, Ziyang Luo, Zhen Ye, Guang Chen, Zhiyong Huang, and Jing Ma. 2025. Adammeme: Adaptively probe the reasoning capacity of multimodal large language models on harmfulness. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, page 4234–4253.

Hang Gao, Chenhao Zhang, Fengge Wu, Changwen Zheng, Junsuo Zhao, and Huaping Liu. 2025. Bootstrapping heterogeneous graph representation learning via large language models: A generalized approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16717–16726.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, Jingji Chen, Jingjia Huang, Kang Lei, Liping Yuan, Lishu Luo, Pengfei Liu, Qinghao Ye, Rui Qian, Shen Yan, and 81 others. 2025. Seed1.5-vl technical report. *CoRR*, abs/2505.07062.

Ming Shan Hee, Roy Ka-Wei Lee, and Wen-Haw Chong. 2022. On explaining multimodal hateful meme detection models. In *Proceedings of the ACM web conference 2022*, pages 3651–3655.

Jianzhao Huang, Hongzhan Lin, Liu Ziyan, Ziyang Luo, Guang Chen, and Jing Ma. 2024. Towards low-resource harmful meme detection with lmm agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2269–2293.

Junhui Ji, Xuanrui Lin, and Usman Naseem. 2024. Capalign: Improving cross modal alignment via informative captioning for harmful meme detection. In *Proceedings of the ACM Web Conference 2024*, pages 4585–4594.

Junhui Ji, Wei Ren, and Usman Naseem. 2023. Identifying creative harmful memes via prompt based approach. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 3868–3872. Association for Computing Machinery.

Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. 2019. Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950*.

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*.

Roy Ka-Wei Lee, Rui Cao, Ziqing Fan, Jing Jiang, and Wen-Haw Chong. 2021. Disentangling hate in online memes. In *Proceedings of the 29th ACM international conference on multimedia*, pages 5138–5147.

Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. 2024a. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Zhengpin Li, Mengzhe Jia, Zheng Wei, and Jian Wang. 2024b. Beyond smoothness: A general optimization framework for graph neural networks with negative laplacian regularization. *Neural Networks*, 180:106704.

Hongzhan Lin, Ziyang Luo, Bo Wang, Ruichao Yang, and Jing Ma. 2024. Goat-bench: Safety insights to large multimodal models through meme-based social abuse. *CoRR*, abs/2401.01523.

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*, pages 4734–4744.

Phillip Lippe, Nithin Holla, Shantanu Chandra, Santhosh Rajamanickam, Georgios Antoniou, Ekaterina Shutova, and Helen Yannakoudakis. 2020. A multimodal framework for the detection of hateful memes. *arXiv preprint arXiv:2012.12871*.

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), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 923–947. Association for Computational Linguistics.

Chaochao Lu, Chen Qian, Guodong Zheng, Hongxing Fan, Hongzhi Gao, Jie Zhang, Jing Shao, Jingyi Deng, Jinlan Fu, Kexin Huang, Kunchang Li, Lijun Li, Limin Wang, Lu Sheng, Meiqi Chen, Ming Zhang, Qibing Ren, Sirui Chen, Tao Gui, and 17 others. 2024. From GPT-4 to gemini and beyond: Assessing the landscape of mllms on generalizability, trustworthiness and causality through four modalities. *CoRR*, abs/2401.15071.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Abdullah Mazhar, Zuhair Hasan Shaik, Aseem Srivastava, Polly Ruhnke, Lavanya Vaddavalli, Sri Keshav Katragadda, Shweta Yadav, and Md Shad Akhtar. 2025. Figurative-cum-commonsense knowledge infusion for multimodal mental health meme classification. In *Proceedings of the ACM on Web Conference 2025*, pages 637–648.

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 23817–23839. Association for Computational Linguistics.

Niklas Muennighoff. 2020. Vilio: State-of-the-art visiolinguistic models applied to hateful memes. *arXiv preprint arXiv:2012.07788*.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems*.

Jorge E. Pérez, Jessica Díaz, Javier García Martin, and Bernardo Tabuenca. 2020. Systematic Literature Reviews in Software Engineering - Enhancement of the Study Selection Process Using Cohen's Kappa Statistic. *J. Syst. Softw.*, 168:110657.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. MOMENTA: A multimodal framework for detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4439–4455. Association for Computational Linguistics.

Minodora Sălcudean. 2020. Visual humor through internet memes (ii) from harmless humour to the discriminatory potential of (anti) memes. case study:"the transgender bathroom debate". *Revista transilvania*.

Bruce Schneier. 1999. Attack trees. *Dr. Dobb's journal*, 24(12):21–29.

Shivam Sharma, Firoj Alam, Md Shad Akhtar, Dimitar Dimitrov, Giovanni Da San Martino, Hamed Firooz, Alon Halevy, Fabrizio Silvestri, Preslav Nakov, and Tanmoy Chakraborty. 2022. Detecting and understanding harmful memes: A survey. *arXiv preprint arXiv:2205.04274*.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (multioff) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020*, pages 32–41. European Language Resources Association (ELRA).

Carlo M Valensise, Alessandra Serra, Alessandro Galeazzi, Gabriele Etta, Matteo Cinelli, and Walter Quattrociocchi. 2021. Entropy and complexity unveil the landscape of memes evolution. *Scientific Reports*, 11(1):20022.

Wenjun Xiong and Robert Lagerström. 2019. Threat Modeling - A Systematic Literature Review. *Comput. Secur.*, 84:53–69.

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 793–803. ACM.

A Appendix

A.1 Detailed Definition of DCG

Type Node. In this part, we predefined seven macro types for the meme classification, i.e., *Nationality, Gender, Religion, Human, Animal, Culture, and Political*. Based on this classification system, we can classify the Internet memes into those that represent what the users want to express.

For each macro type, we ask the MLLM to divide it into subtypes to specifically illustrate the details in these memes. Table 5 shows the corresponding relationship between macro types and the subtypes. In this table, we only present the example subtypes in DCG's type nodes, where the subtypes will be extended when new memes occur.

Table 5: The macro type and example subtypes.

Macro-Type	Example Subtypes in DCG's Type Part
Nationality	Countryhumans Historical Event Parody National Stereotype
Gender	Gender Role Reversal Implicit Misogyny Slangs Performative-Male Symbols Transgender Symbols
Religion	Holiday & Ritual Meme Templates Scripture & Figure Parody Islamic Muslim Symbols Buddhist Symbols
Human	Racism Sterotype Disability Stereotype
Animal	Personified Animal Memes Animal Behavior as Metaphor Mimicry in Nature as Meme Template
Culture	Nonsense Literature Abbreviation Culture Versailles Literature Video Game
Political	Politician as Meme Template Political Figure's Biological Reduction Policy & Event Satire

With the previous classification system, the macro types and subtypes will be mapped to the type node $\mathcal{N}_{\text{type}}$'s three-level information. The prompts P_T for type node generation (Figure 10) will control the subtype extension in the meme reason tree and DCG construction.

Reproduction Nodes. In this part, we have defined the design concept, which is the idea and steps that users convey their harmfulness. We need to reproduce their ideas by observing the visible meme's elements, and the graph represents the MLLM's inner understanding of this meme.

To illustrate the details of the reproduction nodes in the DCG, we have provided an example in Figure 5. In this example, the DCG incorporates four reproduction step nodes and one reproduction goal. The macro type is the "*Gender*" and the subtype is the "*Implicit Misogyny Slangs*".

Representation of Reproduction Steps. To represent the graph in the MLLM's input with plain text together with the logic semantics, we use the logic expression, which is concatenated by the logic symbols ($\wedge, \vee, \neg, \rightarrow$). The example's reproduction steps can be formulated as follows:

$$\mathcal{N}_{\mathrm{M}_1} \wedge (\mathcal{N}_{\mathrm{M}_1} \rightarrow \mathcal{N}_{\mathrm{M}_2} \rightarrow \mathcal{N}_{\mathrm{M}_3}) \rightarrow \mathcal{N}_G \quad (2)$$

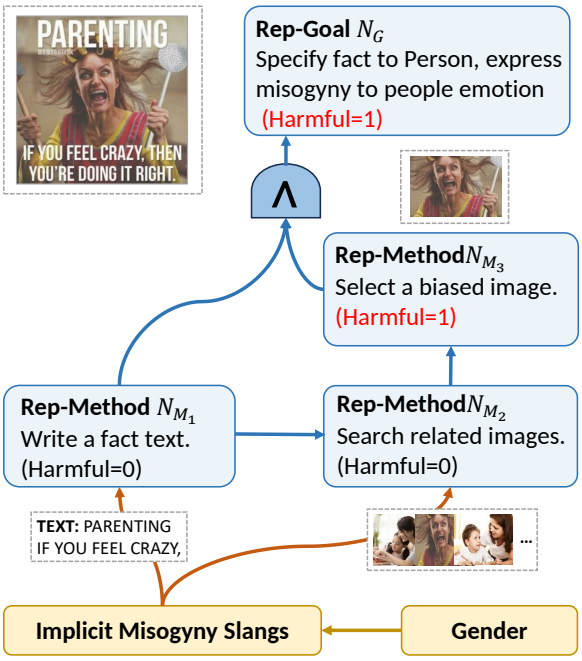

Figure 5: The DCG extracted from the example meme.

where the $\mathcal{N}_{\text{rep}}^{M_1}$ to $\mathcal{N}_{\text{rep}}^{M_3}$ means the content in the reproduction nodes, and the $\mathcal{N}_{\text{rep}}^{G}$ means the content in the reproduction goal. In the guidance of MLLM's inference of Section 3.3, the input logic expression utilizes the "*then*", "*and*", "*or*", and "*on the contrary*" to replace the logic symbols, so the input design concept with the logic expression can be illustrated follows:

Harmful Type
Macro Type: Gender; Subtype: Implicit Misogyny Slangs.
Reproduction Step
The malicious user writes a fact text (no harmful), ***then*** *search the related images based on the fact (no harmful).* ***Then****, it specify the image to the biased people (woman, harmful). The fact* ***and*** *people specification simultaneously cause the attack goal (harmful): Specify fact to person, express misogyny to people emotion.*

A.2 Mathematical Rationality for Algorithm 1

A.2.1 Mathematical Rationality of SVD

We consider a symmetric matrix $\mathbf{A}$ of the form:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{\text{Type}} & \beta\mathbf{E}_{\text{Link}} \\ \beta\mathbf{E}_{\text{Link}} & \alpha\mathbf{A}_{\text{Repr}} \end{pmatrix} \tag{3}$$

where $\mathbf{A}_{\text{Type}}$ and $\mathbf{A}_{\text{Repr}}$ are symmetric matrices representing the adjacency or feature similarity within the type nodes and the reproduction nodes, respectively; $\mathbf{E}_{\text{Link}}$ is the connection matrix between the two parts; and α and β are scaling factors satisfying $1 > \alpha > \beta > 0$.

Effect of Scaling Factors. To show that the scaling factors α and β lead to smaller singular values for the reproduction nodes, we analyze the Rayleigh quotient of $\mathbf{A}$. Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ be a vector partitioned according to the two parts. We can derive the equation:

$$\mathbf{x}^\top\mathbf{A}\mathbf{x} = \mathbf{x}_1^\top\mathbf{A}_{\text{Type}}\mathbf{x}_1 + 2\beta\mathbf{x}_1^\top\mathbf{E}_{\text{Link}}\mathbf{x}_1 + \alpha\mathbf{x}_2^\top\mathbf{A}_{\text{Repr}}\mathbf{x}_2 \tag{4}$$

Consider a vector $\mathbf{x}$ that is nonzero only in the reproduction nodes, i.e., $\mathbf{x} = (\mathbf{0}, \mathbf{x}_2)$ with $|\mathbf{x}_2| = 1$, we can derive the equation:

$$\mathbf{x}^\top\mathbf{A}\mathbf{x} = \alpha\mathbf{x}_2^\top\mathbf{A}_{\text{Repr}}\mathbf{x}_2 \leq \alpha\lambda_{\max}(\mathbf{A}_{\text{Repr}}) \tag{5}$$

where $\lambda_{\max}(\mathbf{A}\text{Repr})$ is the largest eigenvalue of $\mathbf{A}\text{Repr}$. By the Courant-Fischer minimax theorem, the largest singular value of $\mathbf{A}$ (which equals the largest eigenvalue in absolute value, since $\mathbf{A}$ is symmetric) satisfies:

$$\max_{||\mathbf{x}||=1} |\mathbf{x}^\top\mathbf{A}\mathbf{x}| \geq \alpha\lambda_{\max}(\mathbf{A}_{\text{Repr}}) \tag{6}$$

For vectors concentrated on the reproduction nodes, the Rayleigh quotient scales with α, i.e., the singular values associated with the reproduction nodes are roughly proportional to α. Similarly, for vectors that involve both parts, the Rayleigh quotient scales with β. Since $\alpha < 1$ and $\beta < 1$, the scaling reduces the Rayleigh quotient for vectors involving the reproduction nodes, leading to **smaller singular values** for those modes.

Low-rank Approximation and Pruning. In the SVD-based pruning, we approximate $\mathbf{A}$ by a low-rank matrix $\tilde{\mathbf{A}}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\top$, where only the k largest singular values are retained. Since the scaling factors α and β decrease the singular values corresponding to the reproduction nodes, these values are more likely to be truncated in the low-rank approximation. Consequently, the edges within the parts are identified as redundant and can be pruned.

Thus, the scaling strategy with α and β is mathematically justified as it systematically reduces the influence of the reproduction nodes in the singular value spectrum, enabling reproduction nodes' pruning while preserving the type nodes.

A.2.2 Mathematical Rationality for Logarithmic Cut-off Determination

Properties of Long-Tail Distributions. Let $\{x_i\}_{i=1}^n$ be a set of n positive real-valued observations assumed to have a long-tail distribution. After sorting in descending order, we denote: $x_1 \geq x_2 \geq ... \geq x_n > 0$. Our target is to find a threshold x_c such that:

- $x_i \geq x_c$ belongs to the "head" region.
- $x_i < x_c$ belongs to the "tail" region.

The long-tail distributions exhibit the fundamental property that their generating processes are *multiplicative* rather than additive. This leads to the following characterization:

Theorem 1 (Logarithmic Normalization)**.** *For a random variable X following a long-tail distribution, the logarithmic transform $Y = \ln(X)$ typically follows a distribution with exponentially decaying tails (e.g., normal distribution for log-normal, exponential for power-law).*

This theorem converts multiplicative differences in the original scale to **additive** differences in the **logarithmic** scale.

Extreme Value Behavior in Logarithmic Space. Let $Y_1 \geq Y_2 \geq ... \geq Y_n$ be the order statistics of the log-transformed data with a long-tail distribution. The spacings between consecutive order statistics satisfy:

Lemma 1 (Spacing Distribution)**.** *For observations from a distribution with continuous density $f_Y(y)$, the normalized spacings $n \cdot (Y_i - Y_{i+1}) \cdot f_Y(Y_i)$ converge to independent standard exponential random variables as $n \to \infty$.*

This lemma suggests that in regions where the density $f_Y(y)$ is relatively constant, the gaps Δ_i should be approximately equally sized. At the boundary between the head and tail regions, we expect a **discontinuity** in the density, leading to an **anomalously large gap**.

Optimality of Maximum Gap Selection. We provide the theorem for maximum gap optimality:

Theorem 2 (Maximum Gap Optimality)**.** *Assume the data generation follows two regimes:*

- ***Head regime:*** *$Y \sim F_1$ with support on $[a, \infty)$*
- ***Tail regime:*** *$Y \sim F_2$ with support on $(-\infty, b]$*
- *With $a > b$ and minimal overlap between regimes*

As $n \to \infty$, the probability that the maximum gap occurs at the true boundary converges to 1.

Then, we provide the proof for this theorem:

Proof. Let Y_{k^*} and Y_{k^*+1} be the order statistics straddling the true boundary. The expected gap at this position is:

$$\mathbb{E}[\Delta_{k^*}] = \mathbb{E}[Y_{k^*} - Y_{k^*+1}] \geq a - b + o(1) \quad (7)$$

whereas for position $i \neq k^*$, the expected gap is:

$$\mathbb{E}[\Delta_i] = O\left(\frac{1}{n \cdot f_Y(Y_i)}\right)$$

Since $a - b > 0$ represents a fixed separation between regimes, and the other gaps shrink as $O(1/n)$, we have:

$$\lim_{n\to\infty} \mathbb{P}\left(\Delta_{k^*} > \max_{i \neq k^*} \Delta_i\right) = 1$$

Thus, the maximum gap asymptotically identifies the true boundary. □

A.3 Case Study of DCG Construction and Harmful Meme Detection

In this section, we illustrate the case of how the DCG is pruned in the REPMD's second step, including the node extension, score calculation in the SVD, and the details are shown in Figure 6.

In this case, we can see that, REPMD first analyzes the surface-level elements the two historical memes from Figure 1's motivation example. The malicious users incorporate the "*crazy woman*", "*parenting issues*", "*nose stud*", and "*black people*" in these two memes, which are apparently shown in the elements.

Then, REPMD extends the elements to the inner ideas, which are the reproduction steps of design concept that reflect what users have done to design such harmful memes. The two memes are extended to the DCG with 13 nodes (nine reproduction step nodes and four type nodes). From bottom to the top, the DCG represents what the users have done to express misogyny and racism. However, the graph contains too many nodes and edges, which is difficult to understand.

To reduce the DCG's scale, we prune it based on the SVD, which is calculated based on the node's similarity and root correlation. For the node's similarity, we find that the reproduction methods like "*select woman's image*" and "*select black people*" have high TF-IDF similarity, where "*circle the black people's nose*" and the "*specify fact to people*" seem to have low correlation that need to be calibrated and pruned. After SVD pruning and merging, these edges and nodes are removed in the DCG, which only keep the core part that have seven nodes to represent the user's design concept.

A.4 Other Experimental Results

Pruning Time Cost of SVD Replacement. To evaluate the time cost between SVD and MLLMs

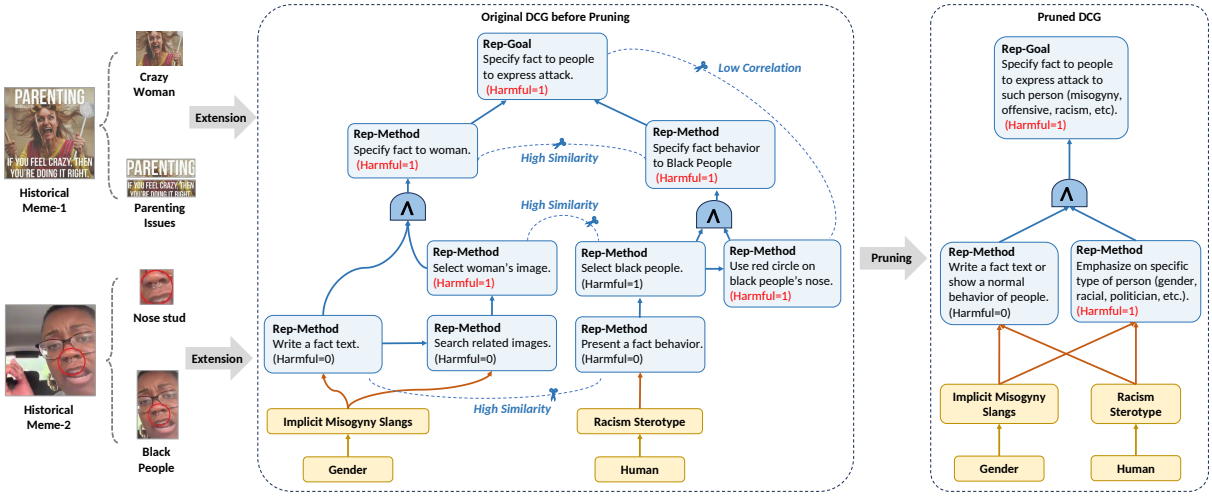

Figure 6: The case of REPMD's DCG pruning.

for graph pruning, we separately evaluate the time the REPMD takes when pruning the historical memes $\{M_1^H, ..., M_n^H\}$. The time cost is calculated with the following equation:

$$Cost_{\text{time}} = Sec(\mathcal{G}_R \rightarrow \mathcal{G}_R^{'})/N(\{M_i^H\}) \quad (8)$$

where the DCG $\mathcal{G}_R$ is prepared based on the historical meme M_i^H's analysis results. With this equation, we do not consider the time cost of graph generation, but only analyze how effectively the graph can be pruned. The tradeoff between the model performance and the efficiency illustrates the benefit of REPMD.

Table 6: The pruning time cost of variants in SVD replacement (seconds per meme for DCG pruning).

Variants	ID	OOD	TE
RepMD w/ SVD	4.9	6.1	5.3
SVD→GPT-4o	62.3 (13×)	61.3 (10×)	52.9 (10×)
SVD→Doubao-V-Pro	*Failed*	71.2 (12×)	69.7 (13×)
SVD→Qwen3VL	*Failed*	*Failed*	75.5 (14×)

Table 6 illustrates the pruning time cost of the SVD and the other MLLMs. We can see that, MLLMs can prune the DCG, but they have too much time cost, which is over 10× to the SVD. In practice, these methods are not applicable, which may come from the fact that the input graph has too many tokens, as well as the long CoT and inference steps that reduce their efficiency. However, we can also see that Qwen3VL-235B can slightly improve the detection performances in the TE task, which means if we can use SVD's results as ground-truth and fine-tune these MLLMs to an acceptable time range (around 2× to 5× to SVD), the MLLMs may be useful to improve REPMD's performance.

Hyperparameter Analysis. We also evaluate the performance of REPMD when we tune the Retained Nodes' Proportion θ from $0\% \sim 100\%$ with a 5% interval. We evaluate the performance of GPT-4o and Doubao-1.5-Vision-Pro on these intervals, where we calculate the **average results** of the OOD and TE experiment.

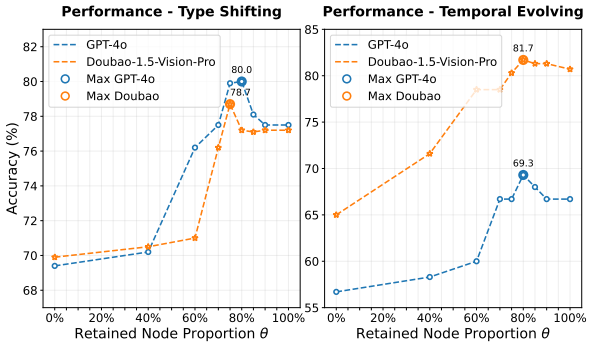

Figure 7: The effect of the retained node's proportion θ.

Figure 7 shows the results of hyperparameter tuning. We can see that, most of the highest value comes from $\theta = 75\%$ and 80%, where the difference between these two parameters is slightly different. We also manually check the values across different types and temporal quarters, and find that the highest results all distribute in the proportion within $70\% \sim 80\%$. Therefore, we choose the balance value for all the experiments, i.e., 75%, for the REPMD's SVD pruning.

A.5 Questionnaire for Human Evaluation

We have provided the content of the questionnaire provided to evaluators, including the introduction of the 5-point Likert scale, the detailed definition for the five criteria, as well as the core dimensions for evaluators to rate our DCG and memes. We also suggest evaluators provide open feedback to our DCG's improvement, so REPMD can be continuously optimized.

Questionnaire: Evaluation of Design Concept Graph (DCG) for Harmful Meme Detection

Instructions: Please evaluate the provided **Design Concept Graph (DCG)** in relation to the corresponding **meme image** based on the following criteria. For each statement, select the response that best matches your assessment using the **5-point Likert scale**, where:
1 = Strongly Disagree
2 = Disagree
3 = Neutral (neither good nor bad / borderline)
4 = Agree
5 = Strongly Agree
Note: A score of **3 represents the threshold between positive and negative evaluation**. Scores above 3 indicate a favorable assessment, while scores below 3 indicate an unfavorable assessment.

Part 1: Criteria Definition
(1) Relevance (R): The DCG is closely related to the content and meaning of the meme image.
(2) Correctness (C): The DCG accurately represents the key elements and message of the meme image.
(3) Actionability (A): The DCG provides clear and practical insights that could guide further actions (e.g., content moderation, design adjustments, or analysis).
(4) Uniqueness (U): The DCG is distinct and differs meaningfully from graphs of other similar meme images.
(5) Explainability (E): The DCG effectively explains how the harmful meme image was designed and reflects the creator's intent or thought process.

Part 2: Core Dimensions

No.	Dimension	Rating (1–5)
1	Relevance (R)	1 2 3 4 5
2	Correctness (C)	1 2 3 4 5
3	Actionability (A)	1 2 3 4 5
4	Uniqueness (U)	1 2 3 4 5
5	Explainability (E)	1 2 3 4 5

Part 3: Open Feedback (Optional) Please provide any additional comments or suggestions regarding the DCG or its evaluation:

Thank you for your participation!

A.6 Case Study of Other Harmful Memes

In Figure 8, we provide two more memes' Qwen2.5VL's output based on our DCG guidance. The output can correct the misleading analysis (e.g., the original racism prediction in case 2) from these memes and output the correct harmful types (e.g., sarcasm of case 2), illustrating its usefulness in harmful meme detection.

A.7 Safeguarding the Text-to-Image Models

After the previous human evaluation that illustrates the benefits of REPMD, we think further on the usefulness of REPMD, i.e., whether the DCG can cheat the Text-to-Image models and output harmful information. Theoretically, if one MLLM cannot correctly detect the harmfulness of the meme, it may be misled by these design concepts without security alignment. Inspired by it, if we can successfully use the text-to-image jailbreak to mislead the MLLM to generate such harmful memes, we may have the chance to protect them based on the design concepts. According to it, we conduct the following two-step experiment.

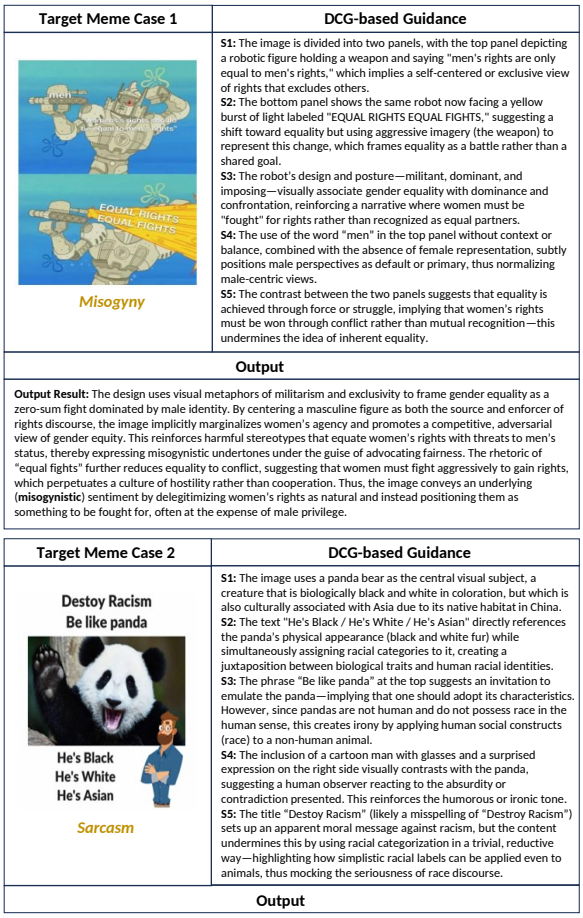

Figure 8: The case study of REPMD of other two meme's Qwen2.5VL's outputs.

Step-1: Attack Prompt Construction: We obtain the target harmful meme M_i^T and its stepwise guidance, then we directly form it as the prompt that will be input to the MLLM:

Task of Qwen-Image
Please draw a meme graph with humor.
Reproduction Step
The malicious user writes a fact text sentence like "parenting is too difficult and makes people crazy", then draws a crazy lady in the image.

Step-2: DCG-Guard Safeguarding: We ask another Qwen3VL-235B model to refer to the DCG, then safeguard the prompt and remove the biased information in the memes, and add the forbidden sentences. The sentences like "*crazy lady in the image*" will be replaced with commands like "***stop generating such image because of misogyny***", or replace them with other benign elements like "***parents that are raising children***".

Original Detection Prompt P_{Harm}

Task of Harmful Meme's Risk Identification
Please identify whether this image contains the risk. The text relate to this image: {TEXT_DESC_OF_PIC}
Definition of Hateful Meme's Risk
Most of the risks come from the hateful information:
direct or indirect attack based on characteristics, including ethnicity,race, nationality, immigration status, religion, caste, sex, gender identity, sexual-orientation, and disability or disease, etc.
Steps to Identify Risks
Step 1: Output what the elements in this image (e.g., image elements, texts, environments, etc.), and what the meaningful information it describes.
Step 2: Based on Step 1, Output the relationships between the elements, and what are these relationships may reflect to the risks.
Step 3: Based on Step 1 and 2, decide whether it contains the risk (e.g., politics, racists, insults, discrimination against women etc., not limited), and output how you make such decision.
Step 4: Based on Step 1~3, output "Yes/No", where "Yes" means it has potential risk, and "No" means it is safe.
Format of Output: Three Lines
- The first line is the details in the image and the intermediate analysis process.
- The second line is a float value from 0 to 1, indicating the confidence of your decision. Lower score means not confident.
  * 0 to 0.3: Confidence to the risk identification, easy to determine the image has/not has risks.
  * 0.3 to 0.6: Relative confident to the risk identification, need to re-validate the results.
  * 0.6 to 0.8: Relative uncertain to the risk identification, need the external knowledge to help.
  * 0.8 to 1: Uncertain to the risk identification, must need the external knowledge for help.
- The second line is the "Yes/No" without any other additional information.

Optimized Detection Prompt P'_{Harm}

Task of Harmful Meme's Risk Identification
Please identify whether this image contains the risk. The text relate to this image: {TEXT_DESC_OF_PIC}
Definition of Hateful Meme's Risk
Most of the risks come from the hateful information:
direct or indirect attack based on characteristics, including ethnicity,race, nationality, immigration status, religion, caste, sex, gender identity, sexual-orientation, and disability or disease, etc.
Steps to Identify Risks
Step 1: Output what the elements in this image (e.g., image elements, texts, environments, etc.), and what the meaningful information it describes.
Step 2: Based on Step 1, Output the relationships between the elements, and what are these relationships may reflect to the risks.
Step 3: Based on Step 1 and 2, decide whether it contains the risk (e.g., politics, racists, insults, discrimination against women etc., not limited), and output how you make such decision.
Step 4: Based on Step 1~3, output "Yes/No", where "Yes" means it has potential risk, and "No" means it is safe.
Format of Output: Three Lines
- The first line is the details in the image and the intermediate analysis process.
- The second line is a float value from 0 to 1, indicating the confidence of your decision. Lower score means not confident.
  * 0 to 0.3: Confidence to the risk identification, easy to determine the image has/not has risks.
  * 0.3 to 0.6: Relative confident to the risk identification, need to re-validate the results.
  * 0.6 to 0.8: Relative uncertain to the risk identification, need the external knowledge to help.
  * 0.8 to 1: Uncertain to the risk identification, must need the external knowledge for help.
- The second line is the "Yes/No" without any other additional information.
Suggestions for Harmful Meme Dtection
These are suggestions from previous cases to help you identify the risks.
Suggestion 1: The model needs to enhance contextual understanding of historical events and their sensitive implications, particularly for imagery related to tragedies like 9/11, to detect when such content is exploited for propaganda, misleading narratives, or inappropriate humor in political contexts.
Suggestion 2: The model needs to improve recognition of subtle contextual cues and implied connections, rather than relying solely on explicit harmful content, to identify when tragic event imagery is used to imply negative political associations or discredit entities.
Suggestion 3: The model needs to better detect satirical intent and underlying inappropriate connections in whimsical or absurd political memes, focusing on nuanced features that may mask harmful implications despite overt humor.
Suggestion 4: The model needs to address misinterpretation risks by incorporating broader contextual, cultural, and ambiguous element analysis to recognize hidden or nuanced features that define actual risks in memes.

Figure 9: The detection prompt P_{Harm} and P'_{Harm}.

We select the Qwen-Image as the target MLLM, which is a model that has not been successfully aligned and may generate harmful meme content, and randomly choose 10∼12 cases as the target memes. We introduce two parts of the evaluation metrics. First, we evaluate how many images are successfully generated based on our attack prompt, and calculate the success rates of the attack prompts. Then, we evaluate the average **SSIM**, which is the metric that represents the similarity between AI-reconstructed images and original images. We evaluate the result with/without DCG-Guard on the five harmful types of memes.

Table 7 shows the results of safeguarding the harmful images through DCG-Guard. We can see that, before safeguarding the Qwen-Image, some attack prompts can achieve nearly 100% success rates (except for the Toxicity memes that contain apparent harmful features in the prompt). With the help of DCG-Guard, the success rates reduce by over 50%, where most of the harmful information in memes is detected and refused to generate with the stop commands. Moreover, even if some images are successfully generated, they may be different from the previous images based on the benign element replacement, with over 0.3 SSIM decrease. These results show that DCG can also help safeguard text-to-image models.

Table 7: The safeguarding of harmful image generation.

Metric	**Generated Img. / Total Guidance** w/o DCG-Guard	w/ DCG-Guard	**Δ**
Racism	7/12 (58.3%)	1/12 (8.3%)	50.0%
Misogyny	8/10 (80.0%)	1/10 (10.0%)	70.0%
Offensiveness	7/10 (70.0%)	1/10 (10.0%)	60.0%
Sarcasm	10/10 (100.0%)	0/10 (0.0%)	100.0%
Toxity	1/10 (10.0%)	0/10 (0.0%)	10.0%
Metric	**Average SSIM** w/o DCG-Guard	w/ DCG-Guard	**Δ**
Racism	0.76	0.12	0.64
Misogyny	0.90	0.33	0.57
Offensiveness	0.54	0.19	0.35
Sarcasm	0.82	-	-
Toxity	0.24	-	-

A.8 Prompts in REPMD

The details of the prompt of harmful meme detection (P_{Harm}), the fail reason tree construction (P_{F}), and DCG construction (P_{D}).

Prompt of Fail Reason Tree Construction P_{F}

Objective:
Given a set of failed meme detection cases (Mfail), you are required to: Generate Reason Nodes (NF): Analyze and explain why the MLLMs failed to detect the harmful content in the meme. Generate Type Nodes (NT): Classify the meme into a macro-type (L1) and further stratify it into subtypes (L2, L3) based on the predefined typology.
Output Format:
Provide clear, structured reasoning for both the failure explanation and the typological classification.
Step 1: Reason Nodes Generation (Mfail → NF)
Input: Concatenated responses from all failed MLLMs for a given meme.
Task: Describe the content of the meme in detail. Analyze and explain why the MLLMs failed to identify the harmful elements. Focus on aspects such as: Subtlety or implicit nature of the harmful content. Cultural or contextual nuances that were overlooked. Limitations in the MLLMs' training data or detection algorithms. Visual or textual ambiguities that led to misinterpretation. Output a succinct, coherent reason node summarizing the failure analysis.
Step 2: Type Nodes Generation ((Mfail, NF) → NT)
Input: The meme description, the generated reason node (NF), and the predefined typology hierarchy.
Task: First, classify the meme into one macro-type (L1) from the following list: Nationality, Gender, Religion, Human, Animal, Culture, Political.
Then, stratify the classification into subtypes: L2: A subcategory under the chosen macro-type. L3: A further refined subcategory under L2 (if applicable).
Ensure the classification is consistent with the content description and the failure analysis. Provide a brief justification for each level of classification.
Predefined Typology Hierarchy (for reference):
Nationality:
L2: Stereotyping, Discrimination, Xenophobia, National Conflict. L3: (e.g., under Stereotyping: Generalization, Caricature, etc.)
Gender:
L2: Sexism, Gender Stereotyping, LGBTQ+ Discrimination, Sexual Harassment. L3: (e.g., under Sexism: Objectification, Role Enforcement, etc.)
Religion:
L2: Blasphemy, Religious Stereotyping, Inter-religious Conflict, Persecution. L3: (e.g., under Blasphemy: Sacred Symbol Misuse, Doctrine Mockery, etc.)
Human:
L2: Dehumanization, Violence, Marginalized Groups, Disability Mockery. L3: (e.g., under Dehumanization: Comparison to Objects, Animalization, etc.)
Animal:
L2: Animal Cruelty, Speciesism, Wildlife Exploitation, Pet Abuse. L3: (e.g., under Animal Cruelty: Graphic Harm, Neglect, etc.)
Culture:
L2: Cultural Appropriation, Tradition Mockery, Ethnic Stereotyping, Heritage Disrespect. L3: (e.g., under Cultural Appropriation: Symbol Misuse, Ritual Trivialization, etc.)
Political:
L2: Propaganda, Hate Speech, Conspiracy Theories, Leader Mockery.
L3: (e.g., under Propaganda: Misinformation, Manipulative Imagery, etc.)
Output Structure:
Meme Description: [Clear description of the meme's content]
Failure Reason (NF): [Concise analysis of why detection failed]
Typology Classification (NT): L1 (Macro-type): [Selected from the predefined list] L2 (Subtype): [Justified subcategory] L3 (Refined Subtype): [Further refinement if applicable]
Justification: [Brief explanation for the classification]
Note
Ensure the analysis is objective, detailed, and grounded in the provided inputs. The classification should align with the harmful elements identified (or missed) in the meme.
Use the generated reason node (NF) to inform the typology classification (NT).

Figure 10: The fail reason tree's construction prompt P_{F}.

Prompt of DCG Node Extension P_{D}

Task Overview
You are Qwen3VL-235B, tasked with extending failure reason nodes (NF) into comprehensive diagnostic graph nodes (GD). This process involves two main phases: Reproduction Method and Logic Gate Extension (analyzing meme construction) Graph Calibration and Goal Extension (validating and contextualizing the analysis)
Input: Failure reason node NF (containing analysis of why detection failed for a specific meme) and the original detection prompt PD
Output: A structured diagnostic graph node GD ready for integration into the overall Diagnostic Causal Graph (DCG)
Phase 1: Reproduction Method and Logic Gate Extension
Step 1.1: Surface-Level Element Extraction
Analyze the meme associated with the failure case NF and identify ALL surface-level elements: Visual elements: Characters, objects, symbols, colors, composition, facial expressions, gestures Textual elements: Captions, labels, speech bubbles, hashtags
Structural elements: Layout, arrangement, visual hierarchy. For each element, describe: Its explicit appearance/representation; Its immediate semantic meaning (literal interpretation)
Combination Logic Nσ Analysis: Describe how these surface elements combine to create meaning. Answer:
logical relationship between elements: and, or, not
Step 1.2: Inner-Level Design Method Analysis (NM)
For EACH identified surface element, conduct a three-question drill-down analysis:
QUESTION A: "Is there a replacement method for this element?"
Consider alternative elements that could serve similar structural or functional roles. Evaluate whether replacement would maintain the meme's format/template; Identify if multiple replacement options exist
QUESTION B: "Why is that element chosen (and not alternatives)?"
Analyze the specific cultural, emotional, or symbolic value of the chosen element. Consider why alternative elements would be less effective for the intended purpose. Examine how this element contributes to the meme's persuasive/rhetorical impact
QUESTION C: "Is the replaced element harmful?"
For each replacement option from Question A, assess whether it would still convey harmful content. If harm persists, describe how the harm mechanism changes or remains. Identify which element properties are essential for conveying harm
Output for Step 1: Complete element inventory with surface descriptions; Combination logic Nσ explanation; For each element: A/B/C analysis results
Phase 2: Graph Calibration and Goal Extension
Step 2.1: Design Concept Validation
Act as a validator (simulating "another Qwen3VL") to assess:
Does the design method analysis (NM) from Phase 1 accurately reflect the actual visual elements?Are there any misinterpretations or overinterpretations of elements? Does the combination logic Nσ correctly describe how the meme functions?
Calibration Checklist: All identified elements actually exist in the meme. Element functions/meanings are not exaggerated or understated. Combination logic reflects actual viewer perception patterns. No significant elements were missed in Phase 1
Step 2.2: User Goal Inference (NG)
Based on the validated analysis, infer what the meme creator likely aimed to achieve:
Primary Goals: What immediate response was sought from viewers? (laughter, anger, agreement, etc.) What belief or attitude was being reinforced or challenged?
Secondary Goals: What social or cultural commentary was being made? What group was being targeted or elevated?
Strategic Goals: Why was this particular format/template chosen? How does the design maximize spread or impact?
Output for Step 2: Calibration report noting any discrepancies or confirmations. Comprehensive user goal analysis NG

Figure 11: The DCG construction's prompt P_{D}.