A Context-Aware Contrastive Learning Framework for Hateful Meme Detection and Segmentation

Xuanyu Su, Yansong Li, Diana Inkpen
University of Ottawa
Ottawa, ON, Canada, K1N 6N5
{xsu072,yli627,diana.inkpen}@uottawa.ca

Nathalie Japkowicz
American University
Washington, DC, USA, 20016-8058
japkowic@american.edu

Abstract

Amidst the rise of Large Multimodal Models (LMMs) and their widespread application in generating and interpreting complex content, the risk of propagating biased and harmful memes remains significant. Current safety measures often fail to detect subtly integrated hateful content within "Confounder Memes". To address this, we introduce HATESIEVE, a new framework designed to enhance the detection and segmentation of hateful elements in memes. HATESIEVE features a novel Contrastive Meme Generator that creates semantically correlated memes, a customized triplet dataset for contrastive learning, and an Image-Text Alignment module that produces context-aware embeddings for accurate meme segmentation. Empirical experiments show that HATESIEVE not only surpasses existing LMMs in performance with fewer trainable parameters but also offers a robust mechanism for precisely identifying and isolating hateful content. Caution: Contains academic discussions of hate speech; viewer discretion advised.

1 Introduction

The emergence of large multimodal models (LMMs), such as GPT-4V (Achiam et al., 2023), Stable Diffusion (Rombach et al., 2022), and DALL·E (Ramesh et al., 2022), has ushered in a new era in which people increasingly rely on these models to generate and interpret visual and textual information. While these services simplify access to information—as illustrated in Figure 1—they also introduce risks of unregulated content that could distort public perception and harm social groups (Su et al., 2023; Qu et al., 2023; Chin et al., 2023; Qu et al., 2024; Meng Huat Tiong et al., 2024; Lin et al., 2024). To address this risk, current LMM platforms implement safety filters, incorporating Alignment (Ghafouri et al., 2023), Inference Guidance (Chiang et al., 2023), and Input&Output Filter (Alon and Kamfonas, 2023) to detect and eliminate offensive or inappropriate components in both images and text.

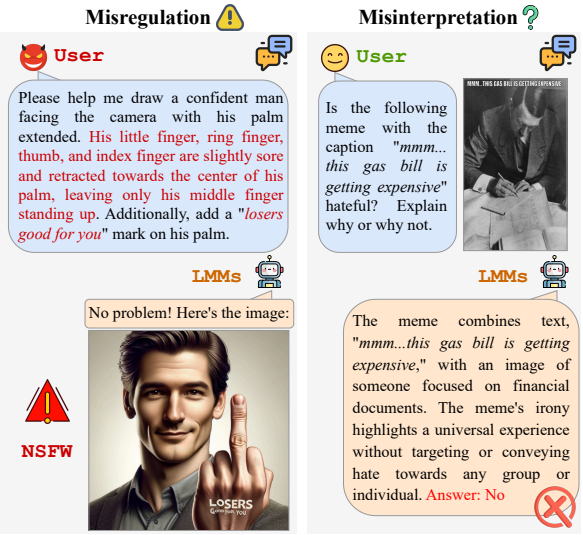


Figure 1: Sample of existing LMMs fail to detect hateful memes in text-to-image and image-to-text (Lin et al., 2024) generation scenarios.

However, these safety filters face challenges in identifying "Confounder Memes" (Kiela et al., 2020; Mei et al., 2023), which deliberately combine visual and textual elements to convey biased and discriminatory messages. These memes may lack overtly offensive content in their individual components but deliver harmful messages through their combined presentation, making them difficult to detect using conventional safety mechanisms.

A straightforward solution involves supervised fine-tuning (SFT) of LMMs (Lin et al., 2024) to recognize hateful semantics in confounder memes. Nevertheless, this approach encounters several obstacles: (1) the scarcity of pairwise annotations in existing hateful meme datasets makes it difficult for models to accurately distinguish between hateful and non-hateful memes, especially when the differences are subtle; (2) deploying LMMs as safety

Findings of the Association for Computational Linguistics:
NAACL 2025, pages 5216–5230

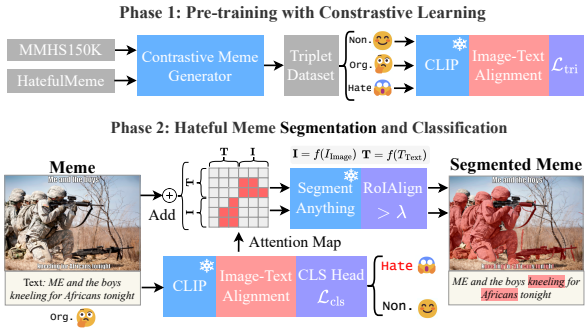


Figure 2: An overview of our HATESIEVE framework. In Phase 1, we use the Contrastive Meme Generator to create a context-correlated triplet meme dataset, pre-training the model via contrastive learning. In Phase 2, we add a classification head and fine-tune the model on the downstream task, enabling it to classify memes while producing segmentation maps of hateful content.

filters alongside their regular online service usage[1] is computationally intensive and non-trivial (Lin et al., 2024). Alternatively, a lightweight classifier (Kumar and Nandakumar, 2022a; Mei et al., 2023) could be trained from scratch using a specialized hateful meme dataset, but this method suffers from limited interpretability and cannot provide detailed segmentation to explain its classifications.

To address these challenges, we introduce HATESIEVE, a novel framework for detecting hateful memes, as detailed in Figure 2. HATESIEVE mitigates the scarcity of detailed annotations by incorporating a **Contrastive Meme Generator** (CMGen), which constructs contextually correlated triplet datasets from existing memes. CMGen generates semantically similar but contrasting hateful and non-hateful memes within the same contextual scenarios, enabling the model to implicitly learn the subtle differences between hateful and non-hateful content. To facilitate detailed meme segmentation, HATESIEVE incorporates an **Image-Text Alignment** (ITA) module coupled with a frozen CLIP model. By pre-training on CMGen-generated triplets using contrastive learning in Phase 1, the ITA module develops context-aware attention maps that effectively segment both image and text hateful elements within memes. In Phase 2, the ITA module incorporates a fine-tuned classification head, leveraging its learned representations for hateful content classification. Empirical experiments conducted on various datasets validate that HATESIEVE not only outperforms existing LMMs with fewer parameters but also excels in interpreting and segmenting the visual and textual components of multimodal memes to effectively identify discriminatory content. Our contributions are summarized as follows:

- We introduce CMGen, which generates context-correlated triplet pairs, filling the gap where specific pairwise annotations are absent in existing hateful meme datasets.
- We present the ITA module that efficiently produces context-aware attention maps for both images and texts. These maps significantly enhance the model's ability to segment and identify discriminatory elements within memes.

2 Related Work

Safety Filter: Existing safety filters for Large Language Models (LLMs) and LMMs typically comprise Alignment (Ghafouri et al., 2023; Touvron et al., 2023; Rafailov et al., 2024; Wu et al., 2024), Inference Guidance (Bai et al., 2022; Chiang et al., 2023; Zhang et al., 2023), and Input&Output Filter components (Alon and Kamfonas, 2023; Hu et al., 2023). Alignment involves fine-tuning LLMs to meet safety objectives using methods such as reinforcement learning from human feedback (RLHF) that optimize models based on safety data and human preferences. Inference guidance steers models towards generating safer responses through system prompts and token selection adjustments during generation. Input&Output filters detect and manage harmful content. However, these methods are primarily designed for unimodal content and struggle to adapt to multimodal content, such as confounder memes.

Alignment necessitates retraining LLMs and massive annotated preference dataset, which is inefficient for online services. Inference guidance depends on LMMs correctly identifying hateful content in memes, which is not always applicable. Additionally, current Input&Output filters generally cater to single modalities, such as the IMSyPP text classification model (Kralj Novak et al., 2022) for text and NSFW filters (Rando et al., 2022) for images in diffusion models. Our HATESIEVE framework addresses these limitations by functioning as an Input&Output filter specifically designed for the meme. It allows to identify and segment both the visual and the textual elements within memes.

[1]"Online service" refers to real-time applications like chatbots, virtual assistants, and image recognition platforms that use LMMs.

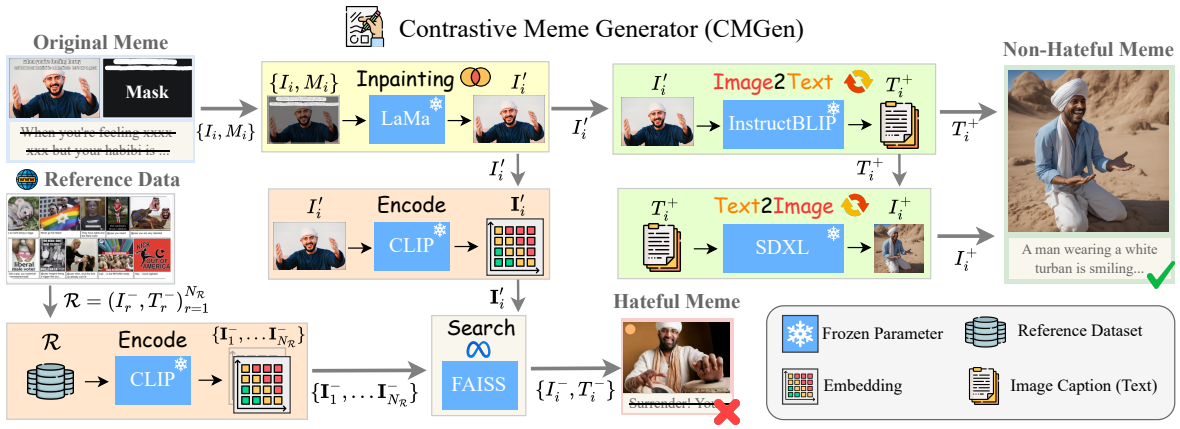


Figure 3: Structure of CMGen: From any meme—including image I_i, text T_i, and caption mask M_i—CMGen generates corresponding hateful and non-hateful counterparts.

Hateful Meme Detection: Current methods for detecting hateful memes generally fall into two categories. The first category, reasoning-based, uses LMMs like LLaVA (Liu et al., 2024) and InstructBLIP (Dai et al., 2024) that generate visual prompts (Li et al., 2023b) based on images. These prompts are concatenated with text data for comprehensive analysis, allowing the LMMs to offer detailed classifications and explanations (Lin et al., 2024). This enables users to assess biases and gain deeper insights into hateful content. However, this approach relies heavily on carefully tailored prompts specifically designed for hate speech detection, making it difficult to create a universal prompt that fits all hateful contexts (Lin et al., 2024). Even minor changes can cause LMMs to misinterpret or overlook hateful memes (Rizwan et al., 2024). While SFT can make LMMs less dependent on prompt design, it is time-consuming and computationally intensive, posing challenges for deployment as safety filters in online services.

Another category of methods uses representation learning and includes lightweight methods such as MOMENTA (Pramanick et al., 2021), PromptHate (Cao et al., 2022), and HateClipper (Kumar and Nandakumar, 2022b). MOMENTA constructs intra-modality attention by integrating external facial recognition data and background knowledge with the CLIP model. PromptHate converts images into text and then classifies them using a language model. HateClipper creates an image-text interaction matrix to fuse multimodal information. These methods enable straightforward classification with fewer parameters, but they offer limited interpretability of their classifications. In contrast, our HATESIEVE framework generates context-aware attention map that enable effective meme segmentation and provide visual interpretation, while delivering classification performance comparable to existing methods.

3 Methodology

The HATESIEVE workflow involves: 1) Generating a triplet dataset with the CMGen. 2) Pre-training the ITA module using the triplet dataset. 3) Extracting attention maps and performing segmentation with the pre-trained ITA. 4) Fine-tuning classification head for hateful content classification.

3.1 Contrastive Meme Generator

As shown in Figure 3, our CMGen is designed to produce both non-hateful and hateful versions of any given meme $\{(I_i, M_i, T_i)\}_{i=1}^{N}$, where $I_i \in \mathbb{R}^{H\times W\times C}$ is the image pixels of the meme, $M_i \in \mathbb{R}^{H\times W}$ is the caption mask, and T_i is the caption overlaid on the meme. These non-hateful and hateful versions are then used for subsequent contrastive learning. The first step in our CMGen is modality separation. By isolating the caption from the meme, we remove text borders and artifacts that may interfere with the image information, ensuring clean image content. Specifically, we apply the LaMA image (Suvorov et al., 2021) inpainting pipeline to extract the pure image content $I_i' = f_{\text{LaMA}}(I_i, M_i)$ from the meme.

To generate the non-hateful version meme (I_i^+, T_i^+), we utilize InstructBLIP (Dai et al., 2024) to create a positive caption $T_i^+ = f_{\text{InstructBLIP}}(I_i')$ of the image content, our prompt is written as follows :"*Please generate a positive and descriptive*

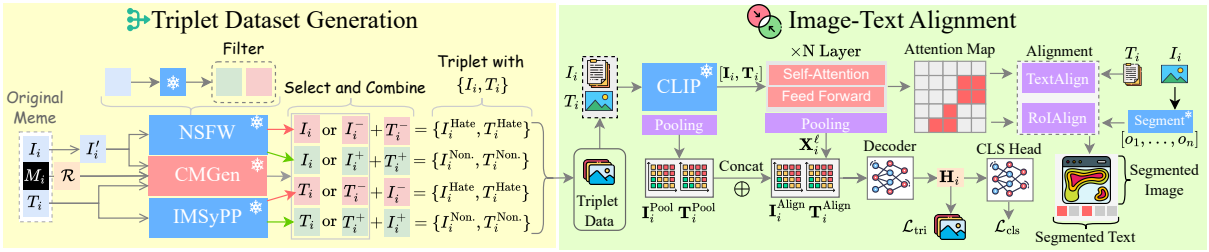


Figure 4: An overview of the triplet dataset generation process and our Image-Text Alignment (ITA) module.

caption for the provided image $\{I_i'\}$." Then, we utilize SDXL with SDEdit (Meng et al., 2021) to produce a high resolution non-hateful image $I_i^+ = f_{\text{SDXL}}(T_i^+)$.

Constructing a hateful version of a meme (I_i^-, T_i^-) presents significant challenges due to the absence of direct annotations regarding ethnic groups, religious affiliations, social groups, or cultural identities in the original meme (I_i, T_i). This lack of explicit metadata complicates the generation of semantically similar hateful memes. To address this issue, we selected the largest available multimodal hate speech dataset, MMHS150k (Gomez et al., 2020), focusing specifically on its "hateful" category to serve as our reference dataset $\mathcal{R} = (I_r^-, T_r^-)_{r=1}^{N_\mathcal{R}}$, where $N_\mathcal{R}$ denotes the number of memes in the reference dataset.

For each purified image I_i' of I_i, we aim to find the most similar hateful image[2] from the reference dataset $\mathcal{R}$. We utilize the CLIP image encoder (Radford et al., 2021) f_{CLIP} to compute the embeddings of both the purified image and the images in the reference dataset. Using FAISS (Douze et al., 2024) for efficient similarity search, we find the index r^* of the most similar image based on Euclidean distance:

$$r^* = \arg\min_{r \in \{1,\ldots,N_R\}} \left\| f_{\text{CLIP}}(I_i') - f_{\text{CLIP}}(I_r^-) \right\|_2$$

The closest hateful pair $(I_{r^*}^-, T_{r^*}^-)$ from the reference dataset is then used as the hateful version of our original meme.

3.2 Triplet Dataset Generation

Our study constructs triplets of meme pairs for contrastive learning, each composed of an original meme (I_i, T_i) and its two variations:

$$\{(I_i, T_i), (I_i^{\text{Non-Hate}}, T_i^{\text{Non-Hate}}), (I_i^{\text{Hate}}, T_i^{\text{Hate}})\}$$

[2]We only use image embeddings for similarity search because the text in memes often lacks explicit social group features, making it less effective for finding semantically similar hateful pairs.

To distinguish between hateful and non-hateful content while maintaining semantic coherence, each meme component—the image I_i and the text T_i—undergoes a pre-filtering process to identify potentially offensive or controversial material. Specifically, each meme is filtered as follows:

- **Text Filtering:** Using the IMSyPP Filter (Kralj Novak et al., 2022), we evaluate the text T_i for offensive or controversial content, assigning a label y_i^T, where $y_i^T = 1$ indicates offensive content and $y_i^T = 0$ indicates non-offensive content.

- **Image Filtering:** Employing the NSFW filter from Stable Diffusion, we assess the image I_i for inappropriate content such as nudity or violence, resulting in a label y_i^I, where $y_i^I = 1$ denotes NSFW content and $y_i^I = 0$ denotes safe content.

As illustrated in Figure 4, we construct the triplet dataset based on these filtering results:

Non-Hateful Pairs $(I_i^{\text{Non-Hate}}, T_i^{\text{Non-Hate}})$**:** We sample from the following combinations to ensure both image and text are non-offensive:

- (I_i^+, T_i^+): The non-hateful meme generated by CMGen without any offensive contents.

- (I_i, T_i^+): The original image $(y_i^I = 0)$ is paired with safe text generated by CMGen.

- (I_i^+, T_i): A safe image generated by CMGen is paired with the original text $(y_i^T = 0)$.

Hateful Pairs $(I_i^{\text{Hate}}, T_i^{\text{Hate}})$**:** We sample from the following combinations to include offensive elements as hateful meme:

- (I_i^-, T_i^-): The hateful meme generated by CMGen that contains offensive content.

- (I_i, T_i^-): The original image ($y_i^I = 1$) is paired with offensive text from CMGen.
- (I_i^-, T_i): An offensive image generated by CMGen is combined with the original text that contains offensive content ($y_i^T = 1$).

3.3 Image-Text Alignment Module

For each meme (I_i, T_i), our ITA module is designed to derive a token/patch-level, context-aware representation that integrates both the image and the text components, as illustrated in Figure 4. The process unfolds as follows:

First, we leverage a pre-trained CLIP encoder to extract initial embeddings for each modality. Specifically, we derive pooled embeddings for text, $\mathbf{T}_i^{\text{Pool}} \in \mathbb{R}^d$, and for images, $\mathbf{I}_i^{\text{Pool}} \in \mathbb{R}^d$, using $f_{\text{CLIP}}(I_i, T_i)$. Additionally, we further extract $\mathbf{T}_i$ and $\mathbf{I}_i$, where $\mathbf{T}_i \in \mathbb{R}^{l \times d}$ and $\mathbf{I}_i \in \mathbb{R}^{o \times d_i}$, using CLIP's text and image encoders, respectively. Here, l represents the text sequence length, o the image patch size, d_i the dimension of the image embedding, and d the dimension of the text embedding.

Then the combined image-text embedding is constructed as $\mathbf{X}_i = [\mathbf{W}_I \mathbf{I}_i, \mathbf{T}_i]$, where $\mathbf{X}_i \in \mathbb{R}^{(o+l) \times d}$ and $\mathbf{W}_I$ is a projection layer designed to map $\mathbf{I}_i$ into the same dimensional space as $\mathbf{T}_i$. To achieve an aligned token-level representation between image and text, we introduce a text-image intra self-attention mechanism, defined as:

$$\text{Attn}_i^\ell = \text{Softmax}\left(\frac{\mathbf{X}_i^\ell \mathbf{W}_{\text{Q}}^\ell (\mathbf{X}_i^\ell \mathbf{W}_{\text{K}}^\ell)^\top}{\sqrt{d_k}}\right) \mathbf{X}_i^\ell \mathbf{W}_{\text{V}}^\ell \tag{1}$$

where d_k is the key dimension, ℓ denotes the layer number, and $\mathbf{W}_{\text{Q}}^\ell, \mathbf{W}_{\text{K}}^\ell, \mathbf{W}_{\text{V}}^\ell$ are the weight matrices for the query, key, and value components in the self-attention layers. The image-text representation is obtained through:

$$\mathbf{X}_i^\ell = f_{\text{Align}}^\ell(\text{Attn}_i^\ell \mathbf{X}_i^{\ell-1}) \tag{2}$$

where f_{Align}^ℓ represents the ℓ-th self-attention block within an L-layer Image-Text Alignment module.

After processing through L layers, the output image-text representation $\mathbf{X}_i^L$ is split and subsequently pooled using the original pooling layer from the CLIP model to form $\mathbf{I}_i^{\text{Align}}$ and $\mathbf{T}_i^{\text{Align}}$. The final image-text representation is then constructed as follows:

$$\mathbf{H}_i = f_{\text{Decoder}}\left([\mathbf{I}_i^{\text{Align}}, \mathbf{T}_i^{\text{Align}}] \oplus [\mathbf{I}_i^{\text{Pool}}, \mathbf{T}_i^{\text{Pool}}]\right) \tag{3}$$

where $\oplus$ denotes the operation for residual connection and f_{Decoder} denotes the decoder, which incorporates a Multilayer Perceptron (MLP) module for dimensionality reduction.

3.4 Training Objective

Our ITA training regimen is organized into two distinct phases: 1) Pre-training through contrastive learning, which equips the ITA module with the ability to effectively segment image and text components within hateful memes, and 2) Fine-tuning for classification tasks, enhancing its ability for specific applications.

Given the generated triplet dataset $\mathcal{D} = \{(I_i, T_i), (I_i^{\text{Non-Hate}}, T_i^{\text{Non-Hate}}), (I_i^{\text{Hate}}, T_i^{\text{Hate}})\}_{i=1}^P$, where P denotes the total number of triplets, we extract the image-text representations for each element in the set as $\{\mathbf{H}_i, \mathbf{H}_i^{\text{Non-Hate}}, \mathbf{H}_i^{\text{Hate}}\}$. For each triplet, where $y_i = 1$ indicates a hateful meme, we identify $\mathbf{H}_i^{\text{Hate}}$ as the positive pair $\mathbf{H}_i^+$ and $\mathbf{H}_i^{\text{Non-Hate}}$ as the negative pair $\mathbf{H}_i^-$. The reverse holds for non-hateful memes with $y_i = 0$. The contrastive learning objective is formulated as follows:

$$\mathcal{L}_{\text{tri}} = \sum_{i=1}^{P} \max\left(0, d(\mathbf{H}_i, \mathbf{H}_i^+) - d(\mathbf{H}_i, \mathbf{H}_i^-) + \epsilon\right)$$

where d represents the Euclidean distance and ϵ is a predefined margin that ensures a minimum discernible difference between the distances of similar and dissimilar pairs.

To adapt the ITA module to the hateful meme classification task, we introduce an additional classification layer f_θ, parameterized by θ, and fine-tune it using the following loss function:

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^{N} \log \mathbb{P}(y_i | \mathbf{H}_i; \theta)$$

where N is the number of examples in the original Hateful Meme dataset.

3.5 Hate Component Segmentation

Our hate component segmentation is structured as follows: After the ITA module is pre-trained via contrastive learning, it can process any given meme (I_i, T_i) to extract a series of self-attention maps $\{\text{Attn}_i\}_{\ell=1}^L$ from all layers. We begin by averaging these self-attention maps across layers to obtain Attn_i'. We then isolate the image attention

map Attn'_{l_j,l_t} and the text attention map[3] Attn'_{l_t,l_j}, where $1 < l_j < L_I + 1$ and $L_I + 1 < l_t < L_T$. Here, l_j denotes the j-th image patch among a total of L_I patches, and l_t indicates the t-th text token within a maximum of L_T text tokens.

Subsequently, we compute the text-aware image attention for each patch:

$$\text{Attn}'_{l_j} = \frac{\sum_{l_t=0}^{L_T} \text{Attn}'_{l_j,l_t}}{L_T}$$

and the image-aware text attention for each text token:

$$\text{Attn}'_{l_t} = \frac{\sum_{l_j=0}^{L_I} \text{Attn}'_{l_t,l_j}}{L_I}$$

To construct an image segmentation map, we employ bilinear interpolation to upscale the $L_I \times L_I$ patch-level attention maps to $H \times W$ pixel-level resolution, facilitating detailed visual analysis of the meme components. As for the text segmentation, we select the Top-k tokens based on the attention scores per token, which allows for precise identification and analysis of the most contextually significant textual elements within the meme. Details of the segmentation process are in Appendix A.1.

4 Experiments

4.1 Setup

Dataset To generate our triplet dataset, we utilized the HatefulMemes (Kiela et al., 2020) and MMHS150k (Gomez et al., 2020) datasets. For contrastive learning training, we incorporated 8,500 entries from the HatefulMemes training set and 33,844 hateful memes sampled from MMHS150k using our contrastive meme generator. For classification fine-tuning, we trained and evaluated our framework's performance on the HatefulMemes *test-unseen* category, as well as on the Harm-C and Harm-P datasets (Pramanick et al., 2021), employing a binary classification setting. Additionally, we assessed the effectiveness of our segmentation approach on the HatefulMemes dataset. Details of the dataset are in Appendix A.2.

Baselines We compare our HATESIEVE framework against the following baseline models for classification task:

- **LMMs:** We evaluate `GPT-4V` (Achiam et al., 2023), `CogVLM` (Wang et al., 2023), `LLaVA-1.5` (Liu et al., 2023), `InstructBLIP` (Dai et al., 2024), `MiniGPT-4` (Zhu et al., 2023), `Qwen-VL` (Bai et al., 2023), `OpenFlamingo` (Awadalla et al., 2023), `MMGPT` (Gong et al., 2023), and `MiniGPT-v2` (Chen et al., 2023) for zero-shot and few-shot (3-shot) inference. Additionally, `LLaVA-1.5`, `InstructBLIP`, and `BLIP2` leverage supervised fine-tuning with QLoRA (Dettmers et al., 2024).

- **CLIP-Based Methods:** We include the original CLIP model as well as its extensions `HateCLIPer` and `MOMENTA`, which build upon CLIP's contrastive embeddings to enhance hateful content detection.

For segmentation tasks, we utilize `InstructBLIP`, `BLIP2`, and CLIP+ITA (a version of HATESIEVE without pre-training). All baseline models are fine-tuned on the HatefulMemes dataset. Detailed segmentation procedures are provided in Appendix A.3.

Metrics For HATESIEVE's classification evaluation, we report Accuracy and F1-score, averaged over five independent runs. Evaluating the segmentation capabilities of HATESIEVE is challenging due to the absence of pixel-level and token-level annotations. To address this, we sampled 100 memes from the HatefulMemes dataset and conducted evaluations using both human annotators and LMMs (Zheng et al., 2024) based on the following criteria:

- **Correctness**: Determines whether the segmentation accurately captures the target social group or elements that reflect the hateful content, based on common-sense understanding.

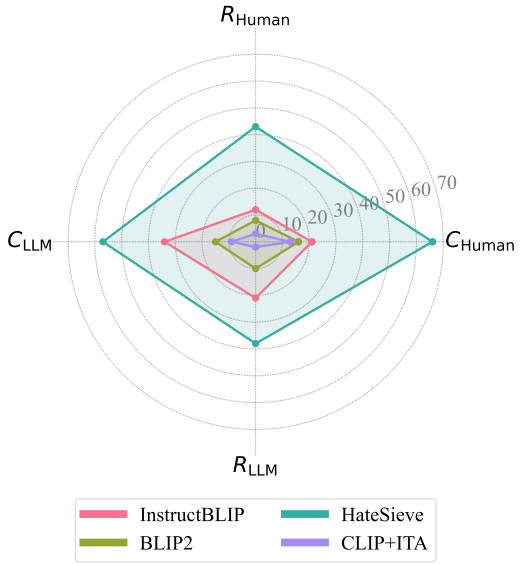
- **Relevance**: Assesses whether the highlighted image segments are meaningfully related to the highlighted text components, ensuring coherence between visual and textual elements.

Each criterion was scored using a binary system: 0 (No) or 1 (Yes). Implementation details for HATESIEVE and the LMM baselines are provided in Appendices A.4 and A.5, respectively. Comprehensive information on the segmentation evaluation process can be found in Appendix A.6.

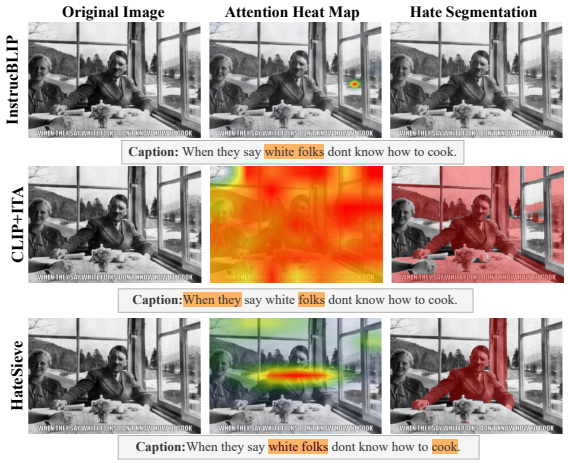
[3]We begin with the second image representation because the first one is a class embedding configured in CLIP, which is not applicable for segmentation purposes.

Model	HatefulMeme Acc.↑	HatefulMeme F1↑	Harm-C Acc.↑	Harm-C F1↑	Harm-P Acc.↑	Harm-P F1↑	# t.p. ↓
Zero-shot Inference							
GPT-4V (-)	71.70	71.28	81.17	80.54	87.42	**88.63**	⊘
CogVLM (17B)	61.50	60.03	57.62	51.38	49.94	44.22	⊘
LLaVA-1.5 (13B)	65.20	61.40	59.15	54.38	56.62	48.77	⊘
InstructBLIP (13B)	58.25	57.42	60.17	36.27	48.19	35.48	⊘
MiniGPT-4 (13B)	58.20	39.98	53.17	48.87	55.55	49.86	⊘
Qwen-VL (10B)	64.00	56.42	56.18	53.94	58.35	52.46	⊘
OpenFlamingo (9B)	58.65	51.78	47.54	43.31	43.69	36.79	⊘
MMGPT (9B)	37.50	27.28	37.16	35.42	33.54	31.97	⊘
MiniGPT-v2 (7B)	57.35	57.27	46.28	42.52	41.37	38.35	⊘
BLIP2 (6.7B)	56.34	55.29	44.37	40.15	39.14	36.59	⊘
Few-shot Learning							
GPT-4V (-)	72.26	71.28	81.16	80.81	87.55	86.07	⊘
LLaVA-1.5 (13B)	65.11	61.68	59.57	54.41	56.72	49.02	⊘
InstructBLIP (13B)	59.12	59.00	62.11	37.17	50.75	35.55	⊘
BLIP2 (6.7B)	57.89	56.93	45.68	41.65	40.58	37.79	⊘
Supervised Fine-Tuning							
InstructBLIP (13B)	63.55	59.34	65.54	42.52	51.98	36.68	65.72M
LLaVA-1.5 (13B)	66.34	63.28	61.61	56.88	59.57	58.62	65.72M
BLIP2 (6.7B)	62.85	56.43	54.28	55.68	45.91	41.37	33.35M
$\text{CLIP}_{\text{Base}}$ (152M)	69.00	62.63	71.88	68.36	65.42	61.08	0.65M
$\text{CLIP}_{\text{Large}}$ (427M)	72.25	68.48	74.23	73.85	80.55	80.25	1.38M
$\text{HateCLIPer}_{\text{Base}}$ (286M)	71.30	68.35	75.31	74.19	81.41	79.68	135.42M
$\text{HateCLIPer}_{\text{Large}}$ (1.5B)	**74.46**	70.15	79.56	77.10	86.89	83.17	1.12B
MOMENTA (434M)	73.34	70.02	**83.82**	82.80	**89.84**	88.26	7.73M
HATESIEVE (155M)	73.45	**71.64**	83.62	**83.07**	88.78	88.53	3.61M

Table 1: Model Performance Comparison. Bold scores indicate the best performance, while underlined scores denote the second-best performance. "Acc." and "F1" represent classification accuracy and macro-F1 score, respectively. "# t.p." denotes the number of trainable parameters.



(a) Comparison of Segmentation Performance: C represents the correctness score, while R indicates the relevance score.



(b) Segmentation effect visualization.

Figure 5: Hateful content segmentation analysis.

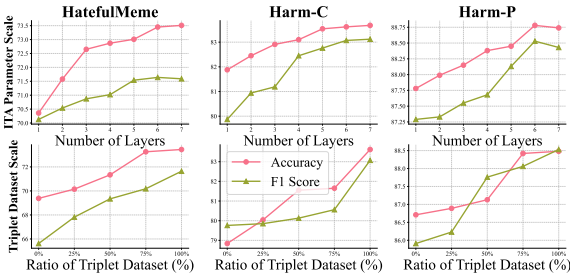
4.2 Classification Results

Table 1 compares the classification performance of various LMMs and CLIP-based methods under zero-shot, few-shot, and supervised fine-tuning (SFT) settings. In the zero-shot scenario, GPT-4V clearly stands out among LMMs, achieving the highest accuracy (71.70%) and F1 score (71.28%) on the HatefulMemes dataset. By contrast, other open-source LMMs (e.g., CogVLM, LLaVA-1.5, and InstructBLIP) show limited capability, with lower accuracies (37.50%–65.20%) and F1 scores (27.28%–71.28%), revealing that pre-training alone is insufficient for capturing the nuanced semantics needed to detect hateful memes.

Under SFT, CLIP-based approaches consistently outperform the LMMs. $\text{HateCLIPer}_{\text{Large}}$ attains the highest accuracy (74.46%) on the HatefulMemes dataset and remains competitive across Harm-C and Harm-P. However, its substantial trainable parameter count (1.12B) raises efficiency concerns for safety filtering applications. In contrast, our proposed HATESIEVE requires only 3.61M trainable parameters, yet achieves the best F1 scores on HatefulMemes (71.64%) and Harm-C (83.07%), and second-best results on Harm-P. These findings underscore the effectiveness of combining contrastive learning pre-training with our ITA module, allowing HATESIEVE to balance strong performance and parameter efficiency while also surpassing GPT-4V in F1 on the HatefulMemes dataset.

4.3 Segmentation Results

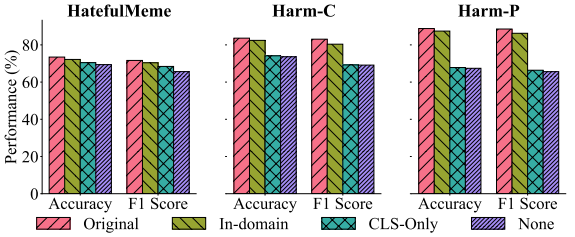
Figure 5a demonstrates that HATESIEVE significantly outperforms InstructBLIP and BLIP2 in both correctness and relevance scores for segmentation, as evaluated by human annotators and LLM evaluators. In contrast, CLIP+ITA—which has not undergone pre-training—underperforms relative to the other models, underscoring the crucial role of contrastive learning pre-training in enhancing hateful content segmentation. Moreover, all models



(a) Impact of ITA Parameter Scale and Triplet Data Scale on Model Performance.

CMGen Strategy	HatefulMeme		Harm-C		Harm-P	
	Acc.	F1	Acc.	F1	Acc.	F1
HATESIEVE	**73.45**	**71.64**	**83.62**	**83.07**	**88.78**	**88.53**
-*w/o* Inpainting	72.61	70.15	82.51	80.13	85.23	84.29
-*w/* Text	71.28	69.43	81.79	81.05	86.38	84.06

(b) Impact of CMGen using different strategies.



(c) Impact of using different pre-training strategies.

Figure 6: Ablation studies on model performance.

achieve slightly lower relevance scores compared to their correctness scores, suggesting that improvements are still needed to more accurately associate specific components within a hateful context. The inter-annotator agreement among human evaluators is discussed in Section A.9.

Figure 5b illustrates the segmentation results, supporting our observations from Figure 5a. Specifically, CLIP+ITA without contrastive learning pre-training generates overly dispersed attention maps. While LMMs effectively identify relevant textual keywords through semantic reasoning, their image segmentation performance suggests that their classification capabilities for hateful memes rely more on the associated Large Language Models rather than on visual information.

5 Ablation Study

ITA Parameter Scale We examined how the number of self-attention layers within the Image-Text Alignment (ITA) module affects the classification performance of HATESIEVE. As shown in Figure 6a, increasing the number of layers initially enhances classification accuracy. However, performance gains plateau and eventually decline when the layer count exceeds six, as evidenced by a noticeable decrease in the F1 score.

Triplet Data Scale We investigated the impact of varying the size of the triplet dataset used during the contrastive learning pre-training stage on the classification performance of HATESIEVE. As illustrated in Figure 6a, we evaluated HATESIEVE's performance when pre-trained with 0% (no pre-training), 25%, 50%, 75%, and 100% of the triplet dataset. The results demonstrate that increasing the amount of pre-training data consistently improves HATESIEVE's classification capabilities.

CMGen Generation Strategy We assessed how text captions influence the quality of the triplet dataset in the CMGen data generation process by evaluating: (1) the role of text captions in matching context-correlated memes based solely on images (-*w/o* inpainting), and (2) the impact of incorporating text embeddings when matching non-hateful pairs using FAISS (-*w/* Text). As shown in Table 6b, residual text captions impair classification performance, indicating interference with image information integration. Additionally, adding text embeddings to FAISS degraded triplet dataset quality, likely due to weak semantic correlations between meme text and images.

Pre-training Strategy We investigated how different pre-training strategies affect the classification performance of HateSieve by comparing three approaches: 1. **In-domain Pre-training**, utilizing only the HatefulMemes training set to directly sample negative image-text pairs without incorporating external reference datasets; 2. **CLS-Only**, replacing the contrastive learning pre-training task with a classification task using the triplet dataset; and 3. **None**, no pre-training. Our results in Figure 6c show that modifying the components of the triplet dataset or altering/removing the pre-training strategy negatively impacts model performance. Notably, adopting the CLS strategy resulted in a decline in performance on the Harm-C and Harm-P datasets that was as significant as no pre-training. This highlights that using classification as a pre-training task doesn't ensure generalizability across various domains.

Transferability of Fine-Tuning Across Datasets We evaluated the transferability of models fine-tuned on one dataset by testing them on different datasets. Table 2 summarizes the results, report-

Training Set	HatefulMemes		Harm-C		Harm-P	
	Acc.	F1	Acc.	F1	Acc.	F1
HM	73.45	71.64	72.54	69.32	72.13	70.02
Harm-C	65.29	63.82	83.62	83.07	80.54	78.52
Harm-P	63.73	61.28	76.44	73.26	**88.78**	**88.53**
Combined (All)	**73.58**	**72.15**	**84.27**	**83.48**	88.52	87.48

Table 2: Transferability of models fine-tuned on one dataset and tested on others. The best performance for each dataset is highlighted in bold.

Category	# Samples	Cls. Acc.	Seg. Acc.	Consis. Rate
Racist	147	69.39	77.55	75.51
Sexist	45	86.67	100.00	86.67
Homophobic	12	50.00	100.00	50.00
Religion	78	73.08	76.92	65.38
Disability	21	71.43	100.00	71.43

Table 3: Results of a manual inspection of 300 memes. *Cls. Acc.* is the rate of correctly identifying memes as hateful or not, *Seg. Acc.* is the rate of correctly segmenting the target entities, and *Consis. Rate* is the proportion of cases where both classification and segmentation are either correct or incorrect.

ing both accuracy and F1 scores for three datasets: HatefulMemes (HM), Harm-C, and Harm-P.

Our findings indicate that models fine-tuned on HM perform best on the HM dataset, but their accuracy and F1 scores drop when evaluated on Harm-C and Harm-P. Conversely, models fine-tuned on Harm-C and Harm-P yield superior performance on their respective datasets, yet underperform on HM. This discrepancy can be attributed to concept drift: HM encompasses a broader range of hate speech categories (including racist, sexist, homophobic, and religious hate), whereas Harm-C and Harm-P predominantly feature memes related to COVID-19 and US politics.

Interestingly, fine-tuning on Harm-C and Harm-P results in less performance degradation when testing on each other's datasets, suggesting that similar content domains exhibit lower concept drift. Moreover, combining all datasets for fine-tuning generally improves performance, highlighting the benefit of diverse and culturally varied training data to enhance generalizability.

Finally, even when fine-tuned on different datasets, our model consistently outperforms most LMMs in both zero-shot and QLoRA settings, demonstrating the robustness and effectiveness of our approach across various evaluation scenarios.

6 Category-Specific Evaluation

We conducted a manual inspection of 300 randomly selected memes from the HatefulMemes dataset, assigning each meme to a hate speech category based on consensus among three annotators. Table 3 presents the classification accuracy (Cls. Acc.), segmentation accuracy (Seg. Acc.), and consistency rate (Consis. Rate) for each category.

While certain categories (e.g., *Sexist*, *Disability*) exhibit high segmentation accuracy, classification accuracy occasionally lags behind (e.g., *Homophobic*). This discrepancy suggests that locating offensive content does not always translate into correct classification. Categories like *Racist* and *Religion* display moderate performance in both metrics, highlighting the need for more diverse training data and targeted refinement. Overall, boosting classification consistency—particularly in underrepresented categories—remains an important goal for future work.

7 Conclusion

We developed HATESIEVE, a framework for classifying and segmenting hateful memes. Our experiments demonstrate that using contrastive learning with a custom triplet dataset enhances classification accuracy and achieves effective segmentation.

Limitations

Our work has several limitations that we plan to address in future research. First, our CMGen system primarily generates context-correlated memes based on image content rather than text, due to inherent restrictions (see Appendix A.7 for a detailed analysis). Second, achieving high accuracy in image segmentation within HATESIEVE remains challenging. Although our current approach uses attention maps at the image-patch level—and we have experimented with refining these maps to pixel-level detail via linear interpolation—this method introduces biases without substantially improving segmentation accuracy. Third, the current version of HATESIEVE focuses exclusively on English hate speech; we plan to extend support to additional languages in future releases. Finally, our framework is not specifically tailored to distinct social or cultural groups, largely due to the limited granularity of annotations in the existing dataset. Future work will concentrate on expanding dataset annotations and enhancing the system's performance across a wider range of multimodal hate speech content.

Ethics Statement

Our research with the Contrastive Meme Generator, which generates both hateful and non-hateful memes, may involve sensitive content. However, all materials are sourced from open-source datasets and confined to academic research, ensuring privacy protection. We adhere to high ethical standards, actively mitigating biases and misuse, and advocate for the responsible use of LMMs.

References

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

Gabriel Alon and Michael Kamfonas. 2023. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Zhi-Yi Chin, Chieh-Ming Jiang, Ching-Chun Huang, Pin-Yu Chen, and Wei-Chen Chiu. 2023. Prompting4debugging: Red-teaming text-to-image diffusion models by finding problematic prompts. *arXiv preprint arXiv:2309.06135*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2024. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv*.

Vahid Ghafouri, Vibhor Agarwal, Yong Zhang, Nishanth Sastry, Jose Such, and Guillermo Suarez-Tangil. 2023. Ai in the gray: Exploring moderation policies in dialogic large language models vs. human answers in controversial topics. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 556–565.

Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. 2020. Exploring hate speech detection in multimodal publications. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1470–1478.

Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. 2023. Multimodal-gpt: A vision and language model for dialogue with humans. *arXiv preprint arXiv:2305.04790*.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Zhengmian Hu, Gang Wu, Saayan Mitra, Ruiyi Zhang, Tong Sun, Heng Huang, and Vishy Swaminathan. 2023. Token-level adversarial prompt detection based on perplexity measures and contextual information. *arXiv preprint arXiv:2311.11509*.

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. *Advances in neural information processing systems*, 33:2611–2624.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.

Petra Kralj Novak, Teresa Scantamburlo, Andraž Pelicon, Matteo Cinelli, Igor Mozetič, and Fabiana Zollo. 2022. Handling disagreement in hate speech modelling. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 681–695. Springer.

Gokul Karthik Kumar and Karthik Nandakumar. 2022a. Hate-clipper: Multimodal hateful meme classification based on cross-modal interaction of clip features. *arXiv preprint arXiv:2210.05916*.

Gokul Karthik Kumar and Karthik Nandakumar. 2022b. Hate-CLIPper: Multimodal hateful meme classification based on cross-modal interaction of CLIP features. In *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, pages 171–183, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Juncheng Li, Kaihang Pan, Zhiqi Ge, Minghe Gao, Wei Ji, Wenqiao Zhang, Tat-Seng Chua, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. 2023a. Finetuning multimodal llms to follow zero-shot demonstrative instructions. In *The Twelfth International Conference on Learning Representations*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

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. *arXiv preprint arXiv:2401.01523*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Jingbiao Mei, Jinghong Chen, Weizhe Lin, Bill Byrne, and Marcus Tomalin. 2023. Improving hateful memes detection via learning hatefulness-aware embedding space through retrieval-guided contrastive learning. *arXiv preprint arXiv:2311.08110*.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.

Anthony Meng Huat Tiong, Junqi Zhao, Boyang Li, Junnan Li, Steven CH Hoi, and Caiming Xiong. 2024. What are we measuring when we evaluate large vision-language models? an analysis of latent factors and biases. *arXiv e-prints*, pages arXiv–2404.

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, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yiting Qu, Xinyue Shen, Xinlei He, Michael Backes, Savvas Zannettou, and Yang Zhang. 2023. Unsafe diffusion: On the generation of unsafe images and hateful memes from text-to-image models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 3403–3417.

Yiting Qu, Xinyue Shen, Yixin Wu, Michael Backes, Savvas Zannettou, and Yang Zhang. 2024. Unsafebench: Benchmarking image safety classifiers on real-world and ai-generated images. *arXiv preprint arXiv:2405.03486*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Javier Rando, Daniel Paleka, David Lindner, Lennart Heim, and Florian Tramèr. 2022. Red-teaming the stable diffusion safety filter. *arXiv preprint arXiv:2210.04610*.

Naquee Rizwan, Paramananda Bhaskar, Mithun Das, Swadhin Satyaprakash Majhi, Punyajoy Saha, and Animesh Mukherjee. 2024. Zero shot vlms for hate meme detection: Are we there yet? *arXiv preprint arXiv:2402.12198*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Xuanyu Su, Yansong Li, Paula Branco, and Diana Inkpen. 2023. Ssl-gan-roberta: A robust semi-supervised model for detecting anti-asian covid-19 hate speech on social media. *Natural Language Engineering*, pages 1–20.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. 2021. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. 2023. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2024. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36.

Zhexin Zhang, Junxiao Yang, Pei Ke, and Minlie Huang. 2023. Defending large language models against jailbreaking attacks through goal prioritization. *arXiv preprint arXiv:2311.09096*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Appendix

A.1 Segmentation Details

To enhance detailed object segmentation, we developed an object highlighting pipeline illustrated in Figure 7. Initially, we extracted the attention map, Attn'_{l_j}, using HATESIEVE and subsequently employed SegmentAnything (Kirillov et al., 2023) to detect and segment objects within the meme. This process produced a series of segmented objects, represented as $O = [o_1, \ldots, o_n]$. We assessed the importance of each object, $\Phi(o_i)$, by integrating the attention map with the object mask using

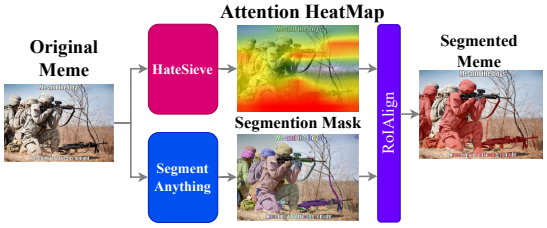


Figure 7: Hate component segmentation process with attention map.

RoIAlign (He et al., 2017). To isolate only the most relevant objects, we implemented a threshold criterion, $\Phi(o_i) > \lambda$, where λ is the pre-established significance threshold.

A.2 Dataset

We utilize several datasets to train and to evaluate the performance of our HATESIEVE framework:

- **HatefulMemes Dataset** (Kiela et al., 2020): Provided by Facebook Research, this dataset comprises 10,000 annotated meme images that combine text and imagery. It is specifically designed to challenge models in detecting hate speech within memes by including subtle and multimodal instances of hateful content.
- **MMHS150k Dataset** (Gomez et al., 2020): This dataset contains 150,000 tweets, each paired with an image, collected between September 2018 and February 2019. The tweets were gathered using 51 Hatebase terms to explore hate speech on social media, offering a rich source of multimodal content for our study.
- **Harm-C and Harm-P Datasets** (Pramanick et al., 2021): Harm-C includes 3,544 memes focusing on COVID-19-related topics, while Harm-P comprises 3,552 memes related to U.S. politics. These datasets provide context-specific challenges for hate speech detection in memes.

By leveraging these diverse datasets, we aim to thoroughly evaluate our model's ability to detect hateful content across different contexts and topics.

A.3 Segmentation with LMMs

To obtain both image and text segmentations using LMMs such as `InstructBLIP` and `BLIP2`, we employ the following prompt:*Please examine the*

Figure 8: Example of segmentation output from the HATESIEVE Framework

provided meme, which includes an [image] and accompanying [text]. Determine if the content can be considered hateful. If you conclude that the meme is hateful, identify and list the specific keywords or phrases in the text.

This prompt enables us to identify the text tokens that `InstructBLIP` considers ambiguous. For image segmentation, we adhere to the approach proposed by Li et al., which involves mapping the query corresponding to the Q-Former in `InstructBLIP` with the image's cross-attention map using bilinear interpolation.

A.4 Implementation Details

Using the Contrastive Meme Generator, we produced a total of 42,344 triplet pairs. During the pre-training and fine-tuning phases, we employed the CLIP-VIT-BASE-PATCH32 as our backbone for the image-text encoder and froze all the CLIP parameters. Our newly introduced Image-Text Alignment module comprises six layers of self-attention blocks. Additionally, we incorporated a two-layer MLP as a decoder for classification fine-tuning.

In the contrastive learning pre-training stage, we used a learning rate of 1e-4 and trained the model over 4 epochs, which took approximately 4 hours on an NVIDIA 4090 GPU. For the fine-tuning stage in the classification task, we fine-tuned the model with a learning rate of 1e-5 for 4 epochs, completing in just 10 minutes. Throughout these stages, the Adam optimizer was utilized, with $\beta = (0.9, 0.999)$.

A.5 LMMs Hyperparameters

For supervised fine-tuning of LMMs, we adopted the QLoRA framework, incorporating trainable parameters ($d = 64$) into the query and key components of the Q-Former. This modification was applied to joint LLM architectures, including OPT-6.7b for `BLIP2` and Vicuna-7b for `InstructBLIP`, while keeping the original parameters frozen. We set a constant dropout rate of 0.05, fixed α at 256, and conducted fine-tuning with a learning rate of 5×10^{-5} and a batch size of 8.

A.6 Segmentation Evaluation

To assess our segmentation module, we employed both human and automated evaluations.

Human Evaluation: Three independent annotators reviewed each segmented meme. Their evaluations were aggregated by majority vote to ensure reliability and minimize bias.

Automated Evaluation with LMMs: We also used GPT-4V as an evaluator, tasking it to score the segmented memes using the same criteria as the human annotators. Both human evaluators and GPT-4V used the evaluation prompt illustrated in Figure 10.

A.7 Triplet Dataset Embedding Analysis

To verify that our CMGen produces context-correlated meme pairs, we conducted an analysis of text and image embedding distances with corresponding positive and negative pairs. We randomly selected 100 pairs from the triplet dataset. As shown in Figure 9, the image embedding distances for both positive and negative pairs (Wasser-

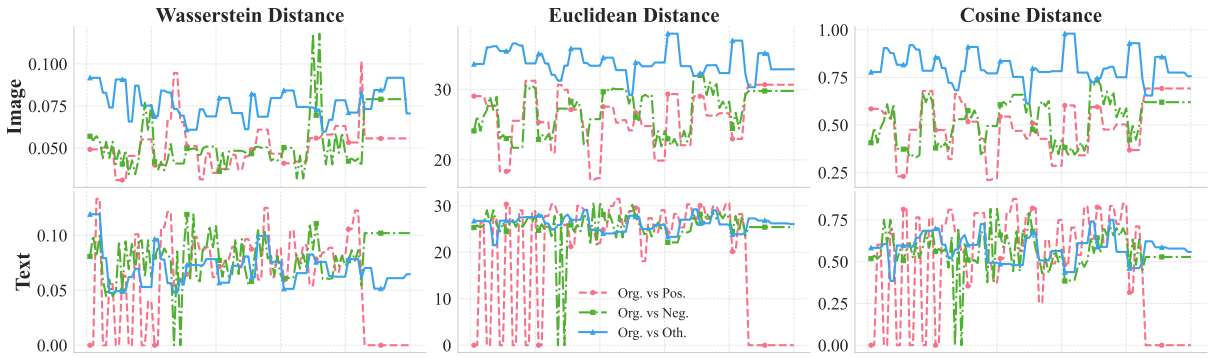


Figure 9: We compare the text and image embeddings of the original meme with its positive (Pos.) and negative (Neg.) pairs, as well as with other randomly selected images and texts (Oth.) from the same dataset, in the triplet dataset using Wasserstein, Euclidean, and Cosine distances. Lower distance values indicate higher similarity, providing a baseline for distance comparison.

stein, Euclidean, and Cosine distances) are consistently lower than the baseline (Others) in most cases, indicating that CMGen successfully generates context-correlated images. However, the text embedding comparison shows that the distances are comparable to the baseline. This is largely because our current CMGen is primarily driven by images, and the text content often lacks detailed information, uses slang, or is challenging to mass-produce with LLMs due to safety policies. We aim to further enhance this aspect of CMGen in the future work.

A.8 Segmentation Results

Additional segmentation results are illustrated in Figure 8. The results demonstrate HATESIEVE's capability to correlate hateful text with objects within images, underscoring the effectiveness of the proposed pre-training with contrastive learning and ITA module.

A.9 Inter-Annotator Agreement

We evaluated inter-annotator agreement among the three annotators by calculating Fleiss' Kappa for both correctness and relevance. Table 4 presents the resulting values along with their interpretations.

Metric	Fleiss' Kappa Score	Interpretation
Correctness	0.7572	Substantial Agreement
Relevance	0.6122	Moderate Agreement

Table 4: Fleiss' Kappa scores for correctness and relevance.

These findings indicate that the annotators achieved substantial agreement on correctness and moderate agreement on relevance. Overall, the results underscore the reliability of our annotation process.

Evaluation Prompt

Given the following segmented meme image $\{I_i\}$ and accompanying text $\{T_i\}$ with highlighted tokens $[x_i, \ldots, x_j]$, please evaluate the segmentation based on the criteria below. For each criterion, assign a score of **0** (No) or **1** (Yes) and provide a brief justification for your decision.

Correctness (Score: 0 or 1):

- Does the segmentation accurately capture the target social group or elements that reflect the hateful content, based on common-sense understanding?
- Consider whether the highlighted areas in the image and text correspond to features commonly associated with the identified hateful content.

Relevance (Score: 0 or 1):

- Are the highlighted image segments meaningfully related to the highlighted text components?
- Assess if the visual elements and the textual tokens work together to convey the intended message, especially in the context of the meme's overall meaning.

Please present your evaluation in the following format:
Correctness: [Score]
Justification: [Your brief explanation]
Relevance: [Score]
Justification: [Your brief explanation]

Figure 10: Evaluation prompt provided to both human annotators and GPT-4V.