Improving Hateful Meme Detection through Retrieval-Guided Contrastive Learning

Anonymous ACL submission

Abstract

Hateful memes have emerged as a significant concern on the Internet. These memes, which are a combination of image and text, often convey messages vastly different from their individual meanings. Detecting hateful memes requires the system to jointly understand the visual and textual modalities. Our investigation reveals that the embedding space of existing CLIP-based systems lacks sensitivity to subtle differences in memes that are vital for correct hatefulness classification. We propose constructing a hatefulness-aware embedding space through retrieval-guided contrastive training. Our approach achieves state-of-the-art performance on the HatefulMemes dataset with an AUROC of 87.0, outperforming much larger fine-tuned Large Multimodal Models like Flamingo and LLaVA. We demonstrate a retrieval-based hateful memes detection system, which is capable of identifying hatefulness based on data unseen in training. This allows developers to update the hateful memes detection system by simply adding new examples without retraining — a desirable feature for real services in the constantly evolving landscape of hateful memes on the Internet.

Disclaimer: This paper contains content for demonstration purposes that may be disturbing to some readers.

1 Introduction

The growth of social media has been accompanied by a surge in hateful content. Hateful memes, which consist of an image accompanied by texts, are becoming a prominent form of online hate speech. This material can perpetuate stereotypes, incite discrimination, and even catalyse real-world violence. To provide users the option of not seeing it, hateful memes detection systems have garnered significant interest in the research community (Kiela et al., 2021; Suryawanshi et al., 2020b,a; Pramanick et al., 2021a; Liu et al., 2022; Hossain et al., 2022; Prakash et al., 2023; Sahin et al., 2023).

Prediction: Benign ✗ Benign ✓ Benign ✓

Figure 1: Illustrative examples from Kiela et al. 2021. The meme on the left is hateful, the middle one is a benign image confounder, and the right one is a benign text confounder. We show HateCLIPper's (Kumar and Nandakumar, 2022) *prediction* below each meme. HateCLIPper misclassifies the hateful meme on the left as benign.

Correctly detecting hateful memes remains difficult. Previous literature has identified a prominent challenge in classifying "confounder memes", in which subtle differences in either image or text may lead to a completely different meaning (Kiela et al., 2021). As shown in Figure 1, the top left and top middle memes share the same caption. However, one of them is hateful and the other benign depending on the accompanying images. Confounder memes resemble real memes on the Internet, where the combined message of images and texts contribute to their hateful nature. Even state-of-the-art models, such as HateCLIPper (Kumar and Nandakumar, 2022), exhibit limited sensitivity to nuanced hateful memes.

We find that a key factor contributing to misclassification is that confounder memes are located in close proximity in the embedding space due to the similarity of text or image content. For instance, HateCLIPper's embedding of the confounder meme in Figure 1 has a high cosine similarity score with the left anchor meme even though they have opposite meanings. This poses challenges for the classifier to distinguish harmful and benign memes.

We propose "**Retrieval-Guided Contrastive**

Learning" (**RGCL**) to learn hatefulness-aware vision and language joint representations. We align the embeddings of same-class examples that are semantically similar with pseudo-gold positive examples and separate the embeddings of opposite-class examples with hard negative examples. We dynamically retrieve these examples during training and train with a contrastive objective in addition to cross-entropy loss. RGCL achieves higher performance than state-of-the-art large multimodal systems on the HatefulMemes dataset with far fewer model parameters. We demonstrate that the RGCL embedding space enables the use of K-nearest-neighbor majority voting classifier. The encoder trained on HarMeme (Pramanick et al., 2021a) can be applied to HatefulMemes (Kiela et al., 2021) without additional training while maintaining high AUC and accuracy using the KNN majority voting classifier, even outperforming the zero-shot performance of large multi-modal models. This allows efficient transfer and update of hateful memes detection systems to handle the fast-evolving landscape of hateful memes in real-life applications. We summarise our contribution as follows:

1. We propose Retrieval-Guided Contrastive Learning for hateful memes detection which learns a hatefulness-aware embedding space via an auxiliary contrastive objective with dynamically retrieved examples. We propose to leverage novel pseudo-gold positive examples to improve the quality of positive examples.

2. Our proposed approach achieves state-of-the-art performance on both the HatefulMemes dataset and the HarMeme dataset, showing its capacity to generalise effectively across various domains of hateful memes.

3. We demonstrate that the retrieval-based KNN majority voting classifier on the learned embedding space outperforms the zero-shot performance of large multimodal models of a much larger scale. This allows easy updating and extension of hateful meme detection systems without retraining.

2 Related Work

Hateful Meme Detection Systems in previous work can be categorised into three types: Object Detection *(OD)-based* vision and language models, *CLIP (Radford et al., 2021) encoder-based* systems, and Large Multimodal Models *(LMM)*.

OD-based models such as VisualBERT (Li et al., 2019), OSCAR (Li et al., 2020), and UNITER (Chen et al., 2020) use Faster R-CNN (Ren et al., 2015) based object detectors (Anderson et al., 2018; Zhang et al., 2021) as the vision model. The use of such object detectors results in high inference latency (Kim et al., 2021).

CLIP-based systems have gained popularity for detecting hateful memes due to their simpler end-to-end architecture. HateCLIPper (Kumar and Nandakumar, 2022) explored different types of modality interaction for CLIP vision and language representations to address challenging hateful memes. In this paper, we show that such CLIP-based models can achieve better performance with our proposed retrieval-guided contrastive learning.

LMMs like Flamingo (Alayrac et al., 2022) and LENS (Berrios et al., 2023) have demonstrated their effectiveness in detecting hateful memes. Flamingo 80B achieves a state-of-the-art AUROC of 86.6, outperforming previous CLIP-based systems although requiring an expensive fine-tuning process.

Contrastive Learning is widely used in vision tasks (Schroff et al., 2015; Song et al., 2016; Harwood et al., 2017; Suh et al., 2019), however, its application to multimodally pre-trained encoders for hateful memes has not been well-explored. Lippe et al. (2020) incorporated negative examples in contrastive learning for detecting hateful memes. However, due to the low quality of randomly sampled negative examples, they observed a degradation in performance. In contrast, our paper shows that by incorporating dynamically sampled positive and negative examples, the system is capable of learning a hatefulness-aware vision and language joint representation.

3 RGCL Methodology

In each training example $\{(I_i, T_i, y_i)\}_{i=1}^{N}$, $I_i \in \mathbb{R}^{C\times H\times W}$ is the image portion of the meme in pixels; T_i is the caption overlaid on the meme; $y_i \in \{0, 1\}$ is the meme label, where 0 stands for benign, 1 for hateful.

We leverage a Vision-Language (VL) encoder to extract image-text joint representations from the image and the overlaid caption:

$$\mathbf{g}_i = \mathcal{F}(I_i, T_i) \quad (1)$$

We encode the training set with our VL encoder to

obtain the encoded retrieval vector database $\mathbf{G}$:

$$\mathbf{G} = \{(\mathbf{g}_i, y_i)\}_{i=1}^{N} \quad (2)$$

We index this retrieval database with Faiss (Johnson et al., 2019) to perform training and retrieval-based KNN classification.

As shown in Figure 2, the VL encoder comprises a frozen CLIP encoder followed by a trainable multilayer perceptron (MLP). The frozen CLIP encoder encodes the text and image into embeddings that are then fused into a joint vision-language embedding before feeding into the MLP.

We use HateCLIPper (Kumar and Nandakumar, 2022) as our frozen CLIP encoder. In Sec.4.4, we compare different choices of the frozen CLIP encoder to demonstrate that our approach does not depend on any particular base model.

3.1 Retrieval Guided Contrastive Learning

For each meme in the training set (the "anchor meme"), we collect three types of contrastive learning examples: (1) pseudo-gold positive; (2) hard negative; (3) in-batch negative to train our proposed retrieval-guided contrastive loss.

(1) Pseudo-gold positive examples are same-label samples in the training set that have high similarity scores under the embedding space. Incorporating these examples pulls same-label memes with similar semantic meanings closer in the embedding space.

(2) Hard negative examples (Schroff et al., 2015) are opposite-label samples in the training set that have high similarity scores under the embedding space. These examples are often confounders of the anchor memes. By incorporating hard negative examples, we enhance the embedding space's ability to distinguish between confounder memes.

(3) For a training sample i, the set of in-batch negative examples (Yih et al., 2011; Henderson et al., 2017) are the examples in the same batch that have a different label as the sample i. In-batch negative examples introduce diverse gradient signals in the training and this causes the randomly selected in-batch negative memes to be pushed apart in the embedding space.

Next, we describe how we obtain these examples to train the system with Retrieval-Guided Contrastive Loss.

3.1.1 Finding pseudo-gold positive examples and hard negative examples

For a training sample i, we obtain the pseudo-gold positive example and hard negative example from the training set with Faiss nearest neighbour search (Johnson et al., 2019) which computes the similarity scores between sample i'th embedding vector $\mathbf{g}_i$ and any target embedding vector $\mathbf{g}_j \in \mathbf{G}$. The encoded retrieval vector database $\mathbf{G}$ is updated after each epoch.

We denote the pseudo-gold positive example's embedding vector:

$$\mathbf{g}_i^{+} = \underset{\mathbf{g}_j \in \mathbf{G}/\mathbf{g}_i}{\operatorname{argmax}} \operatorname{sim}(\mathbf{g}_i, \mathbf{g}_j) \cdot \mathbf{h}(y_i, y_j), \quad (3)$$

$$\mathbf{h}(y_i, y_j) := \begin{cases} 1 & \text{if } y_j = y_i \\ -1 & \text{if } y_j \neq y_i \end{cases}. \quad (4)$$

Similarly for the hard negative example's embedding vector:

$$\mathbf{g}_i^{-} = \underset{\mathbf{g}_j \in \mathbf{G}}{\operatorname{argmax}} \operatorname{sim}(\mathbf{g}_i, \mathbf{g}_j) \cdot (1 - \mathbf{h}(y_i, y_j)). \quad (5)$$

We use cosine similarity for similarity measures.

We denote the embedding vectors for the in-batch negative examples as $\{\mathbf{g}_{i,1}^{-}, \mathbf{g}_{i,2}^{-}, ..., \mathbf{g}_{i,n^-}^{-}\}$. We concatenate the hard negative example with the in-batch negative examples to form the set of negative examples $\mathbf{G}_i^{-} = \{\mathbf{g}_i^{-}, \mathbf{g}_{i,1}^{-}, \mathbf{g}_{i,2}^{-}, ..., \mathbf{g}_{i,n^-}^{-}\}$

3.1.2 RGCL training and inference

Following previous work (Kumar and Nandakumar, 2022; Kiela et al., 2021; Pramanick et al., 2021b), we use logistic regression to perform memes classification as shown in Figure 2. We denote the output from the logistic regression as $\hat{y}_i$ for sample i.

To train the logistic classifier and the MLP within the VL Encoder, we optimise a joint loss function. The loss function consists of our proposed Retrieval-Guided Contrastive Loss (RGCL) and the conventional cross-entropy (CE) loss for logistic regression:

$$\begin{aligned} \mathcal{L}_i &= \mathcal{L}_i^{RGCL} + \mathcal{L}_i^{CE} \\ &= \mathcal{L}_i^{RGCL} + (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)), \end{aligned} \quad (6)$$

where the RGCL loss is computed as:

$$\begin{aligned} \mathcal{L}_i^{RGCL} &= L(\mathbf{g}_i, \mathbf{g}_i^{+}, \mathbf{G}_i^{-}) \\ &= -\log \frac{e^{\operatorname{sim}(\mathbf{g}_i, \mathbf{g}_i^{+})}}{e^{\operatorname{sim}(\mathbf{g}_i, \mathbf{g}_i^{+})} + \sum_{\mathbf{g} \in \mathbf{G}_i^{-}} e^{\operatorname{sim}(\mathbf{g}_i, \mathbf{g})}}. \end{aligned} \quad (7)$$

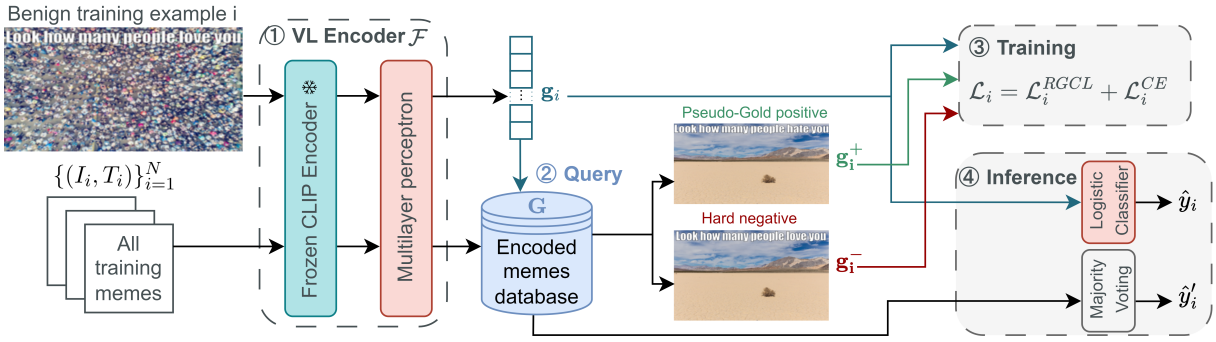

Figure 2: Model overview. (1) Using Vision-Language (VL) Encoder $\mathcal{F}$ to extract the joint vision-language representation for a training example i. Additionally, the VL Encoder encodes the training memes into a retrieval database $\mathbf{G}$. (2) During training, pseudo-gold and hard negative examples are obtained using the Faiss nearest neighbour search. During inference, K nearest neighbours are obtained using the same querying process to perform the KNN-based inference. (3) During training, we optimise the joint loss function $\mathcal{L}$. (4) For inference, we use conventional logistic classifier and our proposed retrieval-based KNN majority voting. For a test meme i, we denote the prediction from logistic regression and KNN classifier as $\hat{y}_i$ and $\hat{y}_i'$, respectively.

3.2 Retrieval-based KNN classifier

We extend our analysis beyond the conventional logistic regression employed in recent models like HateCLIPper. We introduce a retrieval-based KNN majority voting classifier. This majority voting strategy relies on the inherent discrimination capability of the trained joint embedding space. Only when the trained embedding space successfully splits hateful and benign examples will majority voting achieve reasonable performance.

For a test meme t, we retrieve K memes located in close proximity within the embedding space from the retrieval vector database $\mathbf{G}$ (see Eq. 2). We keep a record of the retrieved memes' labels y_k and similarity scores $s_k = \text{sim}(g_k, g_t)$ with the test meme t, where g_t is the embedding vector of the test meme t. We perform similarity-weighted majority voting to obtain the prediction:

$$\hat{y}_t' = \sigma(\sum_{k=1}^{K} \bar{y}_k \cdot s_k), \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function and

$$\bar{y}_k := \begin{cases} 1 & \text{if } y_k = 1 \\ -1 & \text{if } y_k = 0 \end{cases}. \tag{9}$$

We conduct experiments in Sec. 4.2 to show that applying RGCL leads to much better performance with retrieval-based KNN inference than using only the cross-entropy loss.

4 RGCL experiments

We evaluate the performance of the system on the **HatefulMemes** dataset (Kiela et al., 2021) and the **HarMeme** dataset (Pramanick et al., 2021a). The HarMeme dataset consists of COVID-19-related memes collected from Twitter. These memes are labelled with three classes: *very harmful, partially harmful*, and *harmless*. Following previous work (Cao et al., 2022; Pramanick et al., 2021b), we combine the very harmful and partially harmful memes into hateful memes and regard harmless memes as benign memes. The dataset statistics are shown in Appendix C.

To make a fair comparison, we adopt the evaluation metrics commonly used in hateful meme classification (Kumar and Nandakumar, 2022; Cao et al., 2022; Kiela et al., 2021): Area Under the Receiver Operating Characteristic Curve (AUC) and Accuracy (Acc).

We tune the hyperparameters on the development split. We develop our system on the HatefulMemes and use the same hyperparameter settings for training on HarMeme. The experiment setup and hyperparameter settings are detailed in Appendices A and B.

4.1 Comparing RGCL with baseline systems

Table 1 presents the experimental results with logistic regression. Our Retrieval-Guided Contrastive Learning (RGCL) approach is compared to a range of baseline models including Object-detector (OD) based models, Large Multimodal Models (LMM) and CLIP-based systems. On the **HatefulMemes** dataset, RGCL obtains an AUC of 87.0% and an accuracy of 78.8%, outperforming all baseline systems, including the 200 times larger Flamingo-80B.

OD-based models

ERNIE-Vil (Yu et al., 2021), UNITER (Chen et al., 2020) and OSCAR (Li et al., 2020) performs similarly with AUC scores of around 79%.

LMMs

Flamingo-80B (Alayrac et al., 2022) is the previous state-of-the-art model for HatefulMemes, with an AUC of 86.6%. We also fine-tune LLaVA (Liu et al., 2023) (Vicuna-13B Chiang et al., 2023) with the procedure in Appendix D. LLaVA achieves 77.3% accuracy and 85.3% AUC, performing worse than the much larger Flamingo, but better than OD-based models.

CLIP-based systems

PromptHate (Cao et al., 2022) and HateCLIPper (Kumar and Nandakumar, 2022), built on top of CLIP (Radford et al., 2021), outperform both the original CLIP and OD-based models. HateCLIPper achieves an AUC of 85.5%, surpassing the original CLIP (79.8% AUC) but falling short of Flamingo-80B (86.6% AUC). Our system, utilising HateCLIPper's modelling, improves over HateCLIPper by nearly 3% in accuracy, reaching 78.8%. For the AUC score, our system achieves 87.0%, surpassing the previous state-of-the-art Flamingo-80B.

For **HarMeme**, RGCL obtained an accuracy of 87%, outperforming HateCLIPper with an accuracy of 84.8%, PromptHate with an accuracy of 84.5% and LLaVA with an accuracy of 83.3%. Our system's state-of-the-art performance on the HarMeme dataset further emphasises RGCL's robustness and generalisation capacity to different types of hateful memes.

Table 1: Comparing RGCL with baseline systems. Best performance is in **bold**.

Model	HatefulMemes AUC	HatefulMemes Acc.	HarMeme AUC	HarMeme Acc.
Object Detector based models				
ERNIE-Vil	79.7	72.7	-	-
UNITER	79.1	70.5	-	-
OSCAR	78.7	73.4	-	-
Fine-tuned Large Multimodal Models				
Flamingo-80B[1]	86.6	-	-	-
LLaVA (Vicuna-13B)	85.3	77.3	90.8	83.3
Systems based on CLIP				
CLIP	79.8	72.0	82.6	76.7
MOMENTA	69.2	61.3	86.3	80.5
PromptHate	81.5	73.0	90.9	84.5
HateCLIPper[2]	85.5	76.0	89.7	84.8
HateCLIPper **w/ RGCL**	**87.0**	**78.8**	**91.8**	**87.0**

Table 2: Retrieval-based KNN classifier results on HatefulMemes

Model	AUC	Acc.
(I) *Zero shot based on Large Multimodal Models*		
Flamingo-80B	46.4	-
Lens *(Flan-T5 11B)*	59.4	-
InstructBLIP *(Flan-T5 11B)*	54.1	-
InstructBLIP *(Vicuna 13B)*	57.5	-
LLaVA *(Vicuna 13B)*	57.9	54.8
fine-tuned on HarMeme	56.3	54.3
(II) *Train and retrieve on HarMeme*		
HateCLIPper	55.8	51.9
HateCLIPper **w/ RGCL**	**60.0** *(+4.2)*	**57.2** *(+5.3)*
(III) *Train on HarMeme, retrieve on HatefulMemes*		
HateCLIPper	54.4	50.3
HateCLIPper **w/ RGCL**	**66.6** *(+12.2)*	**59.9** *(+9.6)*
(IV) *Train and retrieve on HatefulMemes*		
HateCLIPper	84.6	73.3
HateCLIPper **w/ RGCL**	**86.7** *(+2.1)*	**78.3** *(+5.0)*

4.2 Performance with retrieval-based KNN classifier

Online hate speech is constantly evolving, and it is not practical to keep retraining the detection system. We demonstrate that our system can effectively transfer to the unseen domain of hateful memes without retraining.

We train HateCLIPper with and without RGCL using the HarMeme dataset and evaluate on the HatefulMemes dataset. We report the performance of the KNN classifier when using the HarMeme and HatefulMemes dataset as the retrieval database in Table 2 (II) and (III) respectively. We only use the training set as the retrieval database to avoid label leaking.

We compare our method with state-of-the-art LMMs, including Flamingo (Alayrac et al., 2022), Lens (Berrios et al., 2023), Instruct-BLIP (Ouyang et al., 2022) and LLaVA (Liu et al., 2023) as shown in Table 2 (I). We report the zero-shot performance of these LMMs to replicate the scenario when the model predicts the unseen domain of hateful memes. Furthermore, we report the performance of LLaVA fine-tuned on the HarMeme to align with RGCL's setting in Table 2 (II) and (III).

[1]Flamingo only reports AUC score on HatefulMemes. Since Flamingo is not open-sourced, we are unable to reproduce the accuracy.

[2]HateCLIPper only reports AUC score on HatefulMemes, so we reproduce the system with their released code and obtain the scores.

Lastly, we also report the performance of our methods when trained and evaluated on HatefulMemes in Table 2 (IV).

(**I**) We report LMMs with diverse backbone language models, ranging from Flan-T5 (Chung et al., 2022) and the more recent Vicuna (Chiang et al., 2023). Among these models, Lens with Flan-T5XXL 11B performs the best, achieving an AUC of 59.4%. When LLaVA is fine-tuned on the HarMeme dataset and evaluated on the HatefulMemes dataset, its performance does not improve beyond its zero-shot performance. Its accuracy drops from 54.8% in zero-shot to 54.3% in fine-tuned. These findings indicate that the fine-tuned LLaVA struggles to generalise effectively to diverse domains of hateful memes.

(**II**) When using the **HarMeme** as the retrieval database, our system achieves an AUC of 60.0%, surpassing both the baseline HateCLIPper's AUC of 55.8% and the best LMM's zero-shot AUC score.

(**III**) When using the **HatefulMemes** dataset as the retrieval database, the HateCLIPper's performance degrades, suggesting its embedding space lacks generalising capability to different domains of hateful memes. RGCL boosts the AUC to 66.6%, outperforming the baseline HateCLIPper by a large margin of 12.2% (54.4% for the HateCLIPper). RGCL achieves an accuracy of 59.9%, surpassing the baseline by 9.6% (50.3% for the HateCLIPper). RGCL's AUC and accuracy score also surpass the zero-shot LMMs.

(**IV**) When our system is trained and evaluated on the HatefulMemes dataset (the same system from Table 1), the KNN classifier obtains 86.7% AUC and 78.3% accuracy. These scores also surpass all baseline systems including fine-tuned LMMs in Table 1.

4.3 Effects of incorporating pseudo-gold positive and hard negative examples

In Table 3, we report a comparative analysis by examining performance when specific examples are excluded during the training process.

When we omit the pseudo-gold positive samples, only in-batch positive examples are incorporated during the training. This results in an AUC degradation of 1.0% and accuracy degradation of 1.5%. When the hard negative examples are excluded, leaving only in-batch negative samples, the performance degrades 0.9% and 1.7% for AUC and accuracy, respectively. When removing both types of examples, there is more performance degradation. Both the pseudo-gold positive examples and the hard negative examples are needed for accurately classifying hateful memes.

Table 3: Ablation study on omitting Hard negative and/or Pseudo-Gold positive examples on the HatefulMemes

Model	**AUC**	**Acc.**
Baseline RGCL	**87.0**	**78.8**
w/o Pseudo-Gold positive	86.0	77.3
w/o Hard negative	86.1	77.1
w/o Hard negative and Pseudo-gold positive	85.5	76.8

4.4 Effects of different VL Encoder

We ablate the performance when incorporating RGCL on various VL encoders. As shown in Table 4, we experiment with various encoders in the CLIP family: the original CLIP (Radford et al., 2021), OPENCLIP (Ilharco et al., 2021; Schuhmann et al., 2022; Cherti et al., 2023), and AltCLIP (Chen et al., 2022). Our method boosts the performance of all these variants of CLIP by around 3%.

To verify that our method does not depend on the CLIP architecture, we carry out experiments with ALIGN[3] (Jia et al., 2021). As shown in Table 4, RGCL enhances the AUC score by a margin of 4.4% over the baseline ALIGN model.

Table 4: Ablation study on various Vision-Language Encoder on the HatefulMemes dataset

Model	**AUC**	**Acc.**
HateCLIPper	85.5	76.0
HateCLIPper w/ RGCL	87.0 ***(+1.5)***	78.8 ***(+2.8)***
CLIP	79.8	72.0
CLIP w/ RGCL	83.8 ***(+4.0)***	75.8 ***(+3.8)***
OpenCLIP	82.9	71.7
OpenCLIP w/ RGCL	84.1 ***(+1.2)***	75.1 ***(+3.4)***
AltCLIP	83.4	74.1
AltCLIP w/ RGCL	86.5 ***(+3.1)***	76.8 ***(+2.7)***
ALIGN	73.2	66.8
ALIGN w/ RGCL	77.6 ***(+4.4)***	68.9 ***(+2.1)***

[3]ALIGN only open-sourced the base model which is less capable than the larger CLIP-based models.

4.5 Effects of dense/sparse retrieval

Pseudo-gold positive examples and hard negative examples can be obtained by dense and sparse retrieval during training. To perform sparse retrieval, we carry out image-to-text transformation using object detection. We detail our approach for sparse retrieval in Appendix F.

As shown in Table 5, using a variable number of objects in object detection performs the best in sparse retrieval. The AUC score is comparable with the dense retrieval baseline, however, the accuracy degrades by 0.7%. When using a fixed number of objects in object detection, the performance degrades even more. Using dense retrieval to obtain the pseudo-gold positive examples and hard negative examples achieves better performance.

Table 5: Ablation study of Dense retrieval and Sparse retrieval to obtain pseudo-gold positive examples and hard negative examples on the HatefulMemes dataset

Model	**AUC**	**Acc.**
Baseline *with Dense Retrieval*	87.0	78.8
w/ Variable No. of objects	87.0	78.1
w/ 72 objects	86.1	77.1
w/ 50 objects	85.9	78.6

4.6 Effects of loss function and similarity metrics

Inner product (IP) and Euclidean L2 distance are also commonly used as similarity measures. Since Euclidean distance (L2) is a distance metric, we take its negative to serve as a measure of similarity. We tested these alternatives and found cosine similarity performs slightly better as shown in Table 6.

Additionally, another popular loss function for ranking is triplet loss which compares a positive example with a negative example for an anchor meme. Our results in Table 6 suggest using triplet loss performs comparably to the default NLL loss.

4.7 Qualitative Analysis

We show confounder examples from HatefulMemes in Table 7. In Table 7 (a), both the image and text confounders appear benign. Specifically, the image confounder (middle) presents a meme with the caption "This is the worst cancer I've ever seen," accompanied by an image of two doctors discussing the disease. The text confounder (right) shows a meme praising the flag of Israel with the caption "the flag flies high and proud." However,

Table 6: Ablation study on the loss function and similarity metrics on the HatefulMemes dataset. Similarity metrics include cosine similarity, inner product and negative squared L2.

Loss	Similarity	**AUC**	**Acc.**
NLL	Cosine	**87.0**	**78.8**
	Inner Product	86.1	78.2
	L2	85.7	76.6
Triplet	Cosine	86.7	78.7
	Inner Product	86.1	78.2
	L2	85.7	76.8

when the text and image of these two memes are combined, an extremely hateful and antisemitic meme emerges (left).

HateCLIPper misclassifies the anchor meme (left) as benign with a hateful probability of 0.454. The high cosine similarity scores of the anchor meme with the confounder memes (0.702 and 0.733 respectively) support the notion that these memes, differing in only one modality, are positioned closely in the embedding space. The resulting highly similar joint vision-language embeddings contribute to misclassification.

Our system correctly predicts the anchor meme's hatefulness with a probability of 0.999. Additionally, our system demonstrates very low similarity scores between the anchor meme and the confounder memes (-0.751 and -0.571 respectively).

Table 7 (b) and (c) demonstrate similar cases, where our system separates confounder memes in the embedding space. This implies that our proposed RGCL effectively learns a hatefulness-aware embedding space, placing the meme within the embedding space with a comprehensive hateful understanding derived from both vision and language components.

5 Conclusion

We introduced Retrieval-Guided Contrastive Learning to enhance any VL encoder in addressing challenges in distinguishing confounding memes. Our approach uses novel auxiliary task loss with retrieved examples and significantly improves contextual understanding. Achieving an AUC score of 87.0% on the HatefulMemes dataset, our system outperforms prior state-of-the-art models, including the 200 times larger Flamingo-80B. Our approach also demonstrated state-of-the-art results on the HarMeme dataset, emphasising its usefulness across diverse meme domains.

Table 7: Visualisation for the confounder memes in the HatefulMemes dataset. We present triplets of memes including the hateful anchor memes, the benign image confounders and the benign text confounders. We show the output hateful probability and predictions from HateCLIPper and our RGCL system. We provide the cosine similarity score between the anchor meme and its corresponding confounder meme.

(a)

Ground truth labels	Anchor memes Hateful	Image confounders Benign	Text confounders Benign
Meme			
HateCLIPper			
Probability	0.454	0.000	0.001
Prediction	Benign ✗	Benign	Benign
Similarity with anchor	-	0.702	0.733
HateCLIPper w/ RGCL (Ours)			
Probability	0.999	0.000	0.000
Prediction	**Hateful ✓**	Benign	Benign
Similarity with anchor	-	**-0.751**	**-0.571**

(b)

Meme			
HateCLIPper			
Probability	0.038	0.000	0.001
Prediction	Benign ✗	Benign	Benign
Similarity with anchor	-	0.898	0.913
HateCLIPper w/ RGCL (Ours)			
Probability	1.00	0.000	0.000
Prediction	**Hateful ✓**	Benign	Benign
Similarity with anchor	-	**-0.803**	**-0.769**

(c)

Meme			
HateCLIPper			
Probability	0.385	0.001	0.005
Prediction	Benign ✗	Benign	Benign
Similarity with anchor	-	0.869	0.781
HateCLIPper w/ RGCL (Ours)			
Probability	0.996	0.000	0.000
Prediction	**Hateful ✓**	Benign	Benign
Similarity with anchor	-	**-0.980**	**-0.998**

6 Limitation

Various work defines hate speech differently, and they frequently use other terminology, such as online harassment, online aggression, cyberbullying, or harmful speech. United Nations Strategy and Plan of Action on Hate Speech stated that the definition of hateful could be controversial and disputed (Nderitu, 2020). Additionally, according to the UK's Online Harms White Paper, harms could be insufficiently defined (Woodhouse, 2022). We restrict our definition of hate speech from the two datasets: HatefulMemes (Kiela et al., 2021) and HarMeme (Pramanick et al., 2021a) which may not cover all possible aspects of hate speech. In examining the error cases of our model, we find that the model is unable to recognise subtle facial expressions. This can be improved by using a more powerful vision encoder to enhance image understanding. We leave this to future work.

7 Ethical statement

The HatefulMemes and HarMeme datasets were curated and designed to help fight online hate speech for research purposes only. Throughout the research, we strictly follow the terms of use set by their authors.

Our system is designed to alert social media users of hateful content. However, we recognise an ethical concern: the potential misuse of our system as training signals for image generative models for generating hateful memes. Such misuse could lead to the creation of hateful memes that are wrongly classified as benign by existing detection systems. We emphasise the ethical responsibility when using this system.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 6077–6086.

William Berrios, Gautam Mittal, Tristan Thrush, Douwe Kiela, and Amanpreet Singh. 2023. Towards language models that can see: Computer vision through the lens of natural language. (arXiv:2306.16410). ArXiv:2306.16410 [cs].

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.

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*, volume 12375 of *Lecture Notes in Computer Science*, page 104–120. Springer International Publishing, Cham.

Zhongzhi Chen, Guang Liu, Bo-Wen Zhang, Fulong Ye, Qinghong Yang, and Ledell Wu. 2022. Altclip: Altering the language encoder in clip for extended language capabilities. (arXiv:2211.06679). ArXiv:2211.06679 [cs].

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. 2023. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. (arXiv:2210.11416). ArXiv:2210.11416 [cs].

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. (arXiv:2305.06500). ArXiv:2305.06500 [cs].

Ben Harwood, Vijay Kumar B. G, Gustavo Carneiro, Ian Reid, and Tom Drummond. 2017. Smart mining for deep metric learning. (arXiv:1704.01285). ArXiv:1704.01285 [cs].

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yunhsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. (arXiv:1705.00652). ArXiv:1705.00652 [cs].

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. Mute: A multimodal dataset for detecting hateful memes. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Student Research Workshop*, page 32–39, Online. Association for Computational Linguistics.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip. If you use this software, please cite it as below.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. (arXiv:2102.05918). ArXiv:2102.05918 [cs].

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 6769–6781, Online. Association for Computational Linguistics.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2021. The hateful memes challenge: Detecting hate speech in multimodal memes. (arXiv:2005.04790). ArXiv:2005.04790 [cs].

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning*, page 5583–5594. PMLR.

Gokul Karthik Kumar and Karthik Nandakumar. 2022. 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.

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:1908.03557). ArXiv:1908.03557 [cs].

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. *Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks*, volume 12375 of *Lecture Notes in Computer Science*, page 121–137. Springer International Publishing, Cham.

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:2012.12871). ArXiv:2012.12871 [cs].

Chen Liu, Gregor Geigle, Robin Krebs, and Iryna Gurevych. 2022. Figmemes: A dataset for figurative language identification in politically-opinionated memes. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, page 7069–7086, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. (arXiv:2304.08485). ArXiv:2304.08485 [cs].

Wairimu Nderitu. 2020. United nations strategy and plan of action on hate speech.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Nirmalendu Prakash, Ming Shan Hee, and Roy Ka-Wei Lee. 2023. Totaldefmeme: A multi-attribute meme dataset on total defence in singapore. In *Proceedings of the 14th Conference on ACM Multimedia Systems*, MMSys '23, page 369–375, New York, NY, USA. Association for Computing Machinery.

Shraman Pramanick, Dimitar Dimitrov, Rituparna Mukherjee, Shivam Sharma, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021a. Detecting harmful memes and their targets. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2783–2796, Online. Association for Computational Linguistics.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, page 8748–8763. PMLR.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Umitcan Sahin, Izzet Emre Kucukkaya, Oguzhan Ozcelik, and Cagri Toraman. 2023. Arc-nlp at multimodal hate speech event detection 2023: Multimodal methods boosted by ensemble learning, syntactical and entity features. (arXiv:2307.13829). ArXiv:2307.13829 [cs].

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 815–823. ArXiv:1503.03832 [cs].

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. 2016. Deep metric learning via lifted structured feature embedding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 4004–4012, Las Vegas, NV, USA. IEEE.

Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. 2019. Stochastic class-based hard example mining for deep metric learning. page 7251–7259.

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

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Pranav Verma, Mihael Arcan, John Philip McCrae, and Paul Buitelaar. 2020b. A dataset for troll classification of tamilmemes. In *Proceedings of the WILDRE5– 5th Workshop on Indian Language Data: Resources and Evaluation*, page 7–13, Marseille, France. European Language Resources Association (ELRA).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.

John Woodhouse. 2022. Regulating online harms - uk parliament. *UK Parliament*.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, page 247–256, Portland, Oregon, USA. Association for Computational Linguistics.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. (arXiv:2006.16934). ArXiv:2006.16934 [cs].

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. page 5579–5588.

A Experiment Setup

A work station equipped with NVIDIA RTX 3090 and AMD 5900X was used for the experiments. `PyTorch 2.0.1`, `CUDA 11.8`, and `Python 3.10.12` were used for implementing the experiments. HuggingFace transformer library (Wolf et al., 2019) was used for implementing the pretrained CLIP encoder (Radford et al., 2021). Faiss (Johnson et al., 2019) vector similarity search library with version `faiss-gpu 1.7.2` was used to perform dense retrieval. Sparse retrieval was performed with `rank-bm25 0.2.2` [4]. All the reported metrics were computed by `TorchMetrics 1.0.1`. For LLaVA (Liu et al., 2023), we fine-tuned the model on a system with 4 A100-80GB. The runtime was 4 hours on the HatefulMemes and 3 hours on the HarMeme. All the metrics were reported based on the mean of three runs with different seeds.

[4] https://github.com/dorianbrown/rank_bm25

B Hyperparameter

The default hyperparameter for all the models are shown in Table 8. The modelling hyperparameter is based on HateCLIPper's setting (Kumar and Nandakumar, 2022) for a fair comparison. For vision and language modality fusion, we perform element-wise product between the vision embeddings and language embeddings. This is known as align-fusion in HateCLIPper (Kumar and Nandakumar, 2022). The hyperparameters associated with retrieval-guided contrastive learning are manually tuned with respect to the evaluation metric on the development set. With this configuration of hyperparameter, the number of trainable parameters is about 5 million and training takes around 30 minutes.

Table 8: Default hyperparameter values for the modelling and Retrieval-Guided Contrastive Learning (**RGCL**)

Modelling hyperparameter	Value
Image size	336
Pretrained CLIP model	ViT-L-Patch/14
Projection dimension of MLP	1024
Number of layers in the MLP	3
Optimizer	AdamW
Maximum epochs	30
Batch size	64
Learning rate	0.0001
Weight decay	0.0001
Gradient clip value	0.1
Modality fusion	Element-wise product
RGCL hyperparameter	**Value**
# hard negative examples	1
# pseudo-gold positive examples	1
Similarity metric	Cosine similarity
Loss function	NLL
Top-K for retrieval based inference	10

C Dataset statistics

Table 9 shows the data split for the HatefulMemes and HarMeme dataset. To access the Facebook HatefulMemes dataset, one must follow the license from Facebook[5]. HarMeme is distributed for research purpose only, without a license for commercial use.

D LLaVA experiments

For fine-tuning LLaVA (Liu et al., 2023), we follow the original hyperparameters setting[6] for fine-tuning on downstream tasks. For the prompt format, we follow InstructBLIP (Dai et al., 2023). For computing the AUC and accuracy metrics, we also follow InstructBLIP's procedure.

[5] https://hatefulmemeschallenge.com/#download
[6] https://github.com/haotian-liu/LLaVA

Table 9: Statistical summary of HatefulMemes and HarMeme datasets

Datasets	Train		Test	
	#Benign	#Hate	#Benign	#Hate
HatefulMemes	5450	3050	500	500
HarMeme	1949	1064	230	124

E Ablation study on numbers of retrieved examples

We experiment with using more than one hard negative and pseudo-gold positive gold examples in training.

The inclusion of more than one examples for both types of examples causes the performance to degrade. This phenomenon aligns with recent findings in the literature, as Karpukhin et al. (2020) reported that the incorporation of multiple hard negative examples does not necessarily enhance performance in passage retrieval.

Table 10: Ablation study on omitting and using two Hard negative and/or Pseudo-Gold positive examples on the HatefulMemes

Model	**AUC**	**Acc.**
Baseline RGCL	**87.0**	**78.8**
w/ 2 Hard negative	85.9	77.3
w/ 4 Hard negative	85.7	76.0
w/ 2 Pseudo-Gold positive	86.6	78.5
w/ 4 Pseudo-Gold positive	86.3	77.4

F Sparse retrieval

We use VinVL object detector (Zhang et al., 2021) to obtain the region-of-interest object prediction and its corresponding attributes.

After obtaining these text-based image features, we concatenate these text with the overlaid caption from the meme to perform the sparse retrieval. We use BM-25 (Robertson and Zaragoza, 2009) to perform sparse retrieval. For variable number of object predictions, we set a region-of-interest bounding box detection threshold of 0.2, a minimum of 10 bounding boxes, and a maximum of 100 bounding boxes, consistent with the default settings of the VinVL.