Improving Hateful Meme Detection through Retrieval-Guided Contrastive Learning

Jingbiao Mei, Jinghong Chen, Weizhe Lin, Bill Byrne, Marcus Tomalin
Department of Engineering
University of Cambridge
Cambridge, United Kingdom, CB2 1PZ
{jm2245, jc2124, wl356, wjb31, mt126}@cam.ac.uk

Abstract

Hateful memes have emerged as a significant concern on the Internet. 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. 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.

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

1 Introduction

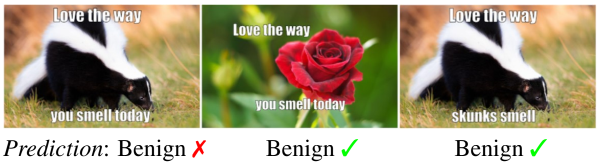


Figure 1: Illustrative examples from Kiela et al. 2021. The meme on the left is hateful, the middle one is a benign image confounder, and the right one is a benign text confounder. We show HateCLIPper's (Kumar and Nandakumar, 2022) *prediction* below each meme. HateCLIPper misclassifies the hateful meme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

The growth of social media has been accompanied by a surge in hateful content. Hateful memes, which consist of images 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).

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

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5333–5347
August 11-16, 2024 ©2024 Association for Computational Linguistics

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 large multi-modal models under similar settings. This allows efficient transfer and update of hateful memes detection systems to handle the fast-evolving landscape of hateful memes in real-life applications. Our contributions are:

1. We propose RGCL 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 HatefulMemes and the HarMeme. We show RGCL's capability across various domains of meme classification tasks on MultiOFF, Harm-P and Memotion7K.

3. Our retrieval-based KNN majority voting classifier facilitates straightforward updates and extensions of hateful meme detection systems across various domains without retraining. With RGCL training, the retrieval-based classifier demonstrates strong cross-dataset generalizability, making it suitable for real services in the dynamic environment of online hateful memes.

2 Related Work

Hateful Meme Detection Systems in previous work can be categorized into three types: Object Detector *(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) and retrieval tasks , however, its application to multimodally pretrained 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.

Sparse retrieval methods, such as BM-25 (Robertson and Zaragoza, 2009) have been used in contrastive learning to obtain collections of hard triplets (Karpukhin et al., 2020; Schroff et al., 2015; Khattab and Zaharia, 2020; Nguyen et al., 2023). In contrast, **dense retrieval**, which is based on vector similarity scores, has been widely adopted for various passage retrieval tasks (Karpukhin et al., 2020; Santhanam et al., 2022; Diaz et al., 2021; Herzig et al., 2021; Lin et al., 2023, 2024). Our method leverages dense retrieval to dynamically select both hard negative and pseudo-gold positive examples.

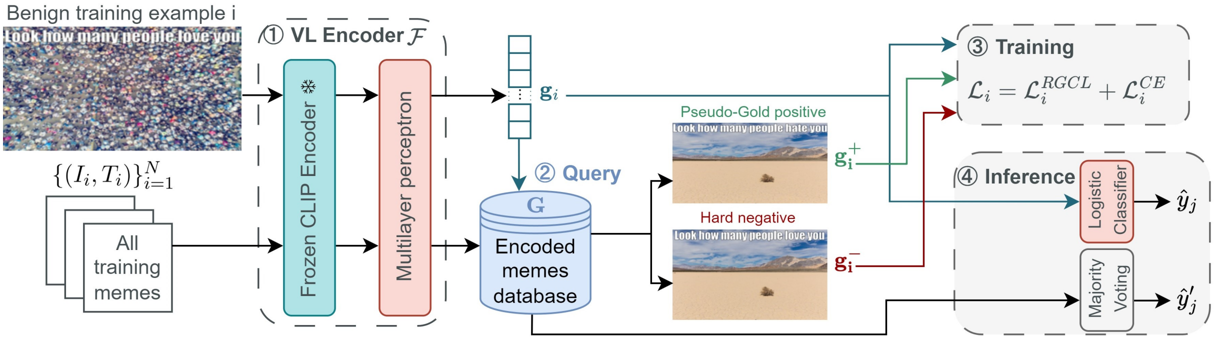


Figure 2: Model overview. ① Using VL Encoder $\mathcal{F}$ to extract the joint vision-language representation for a training example i. Additionally, the VL Encoder encodes the training memes into a retrieval database $\mathbf{G}$. ② During training, pseudo-gold and hard negative examples are obtained 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 j, we denote the prediction from logistic regression and KNN classifier as $\hat{y}_j$ and $\hat{y}'_j$, respectively.

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:

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

We encode the training set with our VL encoder to obtain the encoded retrieval vector database $\mathbf{G}$:

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

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 VL embedding before feeding into the MLP.

We use HateCLIPper (Kumar and Nandakumar, 2022) as our frozen CLIP encoder. The model architecture is detailed in Appendix C. 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 dynamically obtain 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:

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

similarly for the hard negative example's embedding vector:

$$\mathbf{g}_i^- = \underset{\mathbf{g}_j \in \mathbf{G}, y_i \neq y_j}{\operatorname{argmax}} \operatorname{sim}(\mathbf{g}_i, \mathbf{g}_j). \tag{4}$$

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 $\hat{y}_j$ for sample j.

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

$$\begin{aligned}\mathcal{L}_i &= \mathcal{L}_i^{RGCLL} + \mathcal{L}_i^{CE} \\ &= \mathcal{L}_i^{RGCLL} + (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)),\end{aligned} \tag{5}$$

where the *RGCLL* is computed as:

$$\begin{aligned}\mathcal{L}_i^{RGCLL} &= L(\mathbf{g}_i, \mathbf{g}_i^+, \mathbf{G}_i^-) \\ &= -\log \frac{e^{\operatorname{sim}(\mathbf{g}_i, \mathbf{g}_i^+)}}{e^{\operatorname{sim}(\mathbf{g}_i, \mathbf{g}_i^+)} + \sum_{\mathbf{g} \in \mathbf{G}_i^-} e^{\operatorname{sim}(\mathbf{g}_i, \mathbf{g})}}.\end{aligned} \tag{6}$$

In Appendix G, we compare different similarity metrics and loss functions.

3.2 Retrieval-based KNN classifier

In addition to logistic classifier, we introduce a retrieval-based KNN majority voting classifier which 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. The KNN classifier is suitable for real services in the constantly evolving landscape of online hateful memes as the the retrieval database can be extended without the need to retrain the system. In Section 4.2, we show that our proposed KNN classifier generalizes well to unseen data without additional training.

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 $s_k = \operatorname{sim}(g_k, g_t)$ 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:

$$\hat{y}_t' = \sigma(\sum_{k=1}^{K} \bar{y}_k \cdot s_k), \tag{7}$$

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

$$\bar{y}_k := \begin{cases} 1 & \text{if } y_k = 1 \\ -1 & \text{if } y_k = 0 \end{cases}. \tag{8}$$

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 primarily evaluate the performance of RGCL on the **HatefulMemes** dataset (Kiela et al., 2021) and the **HarMeme** dataset (Pramanick et al., 2021a). The HarMeme dataset consists of COVID-19-related harmful memes collected from Twitter. In Section 4.7, we evaluate three additional datasets to show the generalizability of RGCL beyond hateful meme classification. The dataset statistics are shown in Appendix D.

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

The experiment setup, including the statistical significance tests, 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. RGCL is compared to a range of baseline models including OD-based models, LMMs, 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) with the procedure in Appendix E. 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.

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.

	HatefulMemes		**HarMeme**	
Model	**AUC**	**Acc.**	**AUC**	**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 1: Comparing RGCL with baseline systems. Best performance is in **bold**.

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. To ensure a fair comparison, we report the performance of LLaVA fine-tuned on the HarMeme to align with RGCL's setting in Table 2 (II) and (III).

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-

[1] Since Flamingo is not open-sourced, we are unable to obtain accuracy.

[2] Reproduced with HateCLIPper's code base.

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
LR instead of KNN	*52.4*	*49.5*
HateCLIPper **w/ RGCL**	**60.0** *(+4.2)*	**57.2** *(+5.3)*
LR instead of KNN	*59.4 (+7.0)*	*50.9 (+1.4)*
*(**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)*

Table 2: Retrieval-based KNN classifier results on HatefulMemes. LR refers to logistic regression.

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.

Additionally, we provide the results of using logistic regression (LR) as an alternative to the KNN classifier, both with and without RGCL training, when systems trained on HarMeme are tested on HatefulMemes. The performance of logistic regression consistently falls short of the KNN classifier. Logistic regression with RGCL training achieves an AUC of 59.4%, outperforming the HateCLIPper's baseline by 7%. Note that the logistic regression does not the retrieval of examples.

(**III**) When using **HatefulMemes** as the retrieval database, the HateCLIPper's performance degrades, suggesting its embedding space lacks generalizing capability to different domains of hateful memes. RGCL boosts the AUC to 66.6%, outperforming the baseline by a large margin of 12.2%. RGCL achieves an accuracy of 59.9%, surpassing the baseline by 9.6%. 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 examples, only in-batch positive examples are incorporated during the training. This results in an accuracy degradation of 1.5%. Hard positive examples, same-label samples with high similarity scores, are commonly used in contrastive learning literature. In our case, when incorporating hard positive examples rather than pseudo-gold positive examples, the training becomes unstable and results in divergence.

When the hard negative examples are excluded, leaving only in-batch negative samples, the performance degrades 1.7% for accuracy. 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.

When excluding the in-batch negative examples, training becomes unstable and fail to converge, which is consistent with previous findings in (Henderson et al., 2017).

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

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

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.

Model	**AUC**	**Acc.**
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)*

Table 4: Ablation study on various VL encoders on the HatefulMemes dataset

4.5 Effects of dense/sparse retrieval

We compare the commonly used sparse retrieval to our proposed dynamic dense retrieval for obtaining contrastive learning examples. We detail our approach for sparse retrieval in Appendix H.

As shown in Table 5, using a variable number of objects in object detection performs the best in sparse retrieval. However, the accuracy degrades by 0.7% compared to the dense retrieval baseline. When using a fixed number of objects in object detection, the performance degrades even more. Our proposed dynamic dense retrieval obtains better performance than the commonly used sparse retrieval methods.

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

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

4.6 Effects of Retrieval-Guided Contrastive Learning Loss

As shown in Eq. 5, the mixing ratio between RGCLL and the CE loss is 1:1 by default. In Table 6, we compare the different mixing ratios between the two loss functions. We observe a significant performance improvement whenever RGCLL is included. For simplicity, we maintain a 1:1 mixing ratio. Notably, in the absence of cross-entropy loss, we identified several examples where models with RGCL fail but models without RGCL succeed. Conversely, inclusion of cross-entropy loss eliminates such discrepancies.

RGCLL:CE	**Acc.**	**AUC**
0:1	76.0	85.5
0.5:1	78.5	86.8
1:1	78.8	**87.0**
2:1	**79.1**	86.9
4:1	78.6	86.9
1:0	79.0	86.5

Table 6: Ablation study of different mixing ratios for the two type of loss functions on the HatefulMemes dataset

4.7 Effects of RGCL on different Meme Classification tasks

To demonstrate RGCL's versatility beyond hateful meme classification, we assess its efficacy on three additional datasets: MultiOFF (Suryawanshi et al., 2020a), Harm-P (Pramanick et al., 2021b), and Memotion7K (Sharma et al., 2020). These datasets originally used the F1 score as their evaluation metric; we also include Accuracy. We train a separate model for each of the datasets following the procedures detailed in Appendices A and B. Table 7 shows the results for CLIP with and without RGCL training on these three datasets[4].

MultiOFF contains memes related to the 2016 U.S. presidential election sourced from social media sites, such as Twitter and Instagram. The memes are labeled as non-offensive and offensive. MultiOFF is a relatively small dataset, containing less than 500 training examples. RGCL outperforms the baseline by a significant margin of 4.7% in accuracy. RGCL yields consistent gain even with relatively small datasets.

Harm-P contains harmful and harmless memes on US politics sourced from social media sites. RGCL shows more than 2% gain in both accuracy and F1 score over the baseline system.

Memotion7K, designed for multi-task meme emotion analysis, includes annotations for humor, sarcasm, offensiveness, and motivation. RGCL

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

[4] Since CLIP surpasses almost all other prior published systems on these datasets, we do not include prior results in the comparison.

shows improvement over baseline across all four emotion classification tasks with an average gain of more than 3% on both accuracy and F1 scores. These results highlight RGCL's capability for improving emotion detection.

	w/o RGCL		w/ RGCL	
Dataset	Acc.	F1	Acc.	F1
MultiOFF	62.4	54.8	67.1 *(+4.7)*	58.1 *(+3.3)*
Harm-P	87.6	86.9	89.9 *(+2.3)*	89.5 *(+2.6)*
Memotion7K				
-Humour	73.0	83.8	76.3 *(+3.3)*	86.6 *(+2.4)*
-Sarcasm	75.1	85.6	77.3 *(+2.2)*	87.2 *(+1.6)*
-Offensive	72.8	83.5	77.6 *(+4.8)*	87.4 *(+3.9)*
-Motivation	59.6	72.6	62.4 *(+2.8)*	76.8 *(+4.2)*
Average	70.1	81.4	73.4 *(+3.3)*	84.5 *(+3.1)*

Table 7: The performance of CLIP with and without RGCL training on different meme classification tasks

5 Case Analysis

We now analyze how RGCL improves relative to baseline systems on confounding memes.

5.1 Quantitative analysis

From the 500 validation samples of HatefulMemes, we annotated 101 examples and picked 24 confounder memes. On this confounder subset, Hate-CLIPper without RGCL obtains an accuracy of 66.7%, while RGCL significantly boosts the accuracy to 83.3%. These results show that RGCL improves the classification of challenging confounder memes, which exhibit differences in either the image or text.

Next, we analyze how RGCL improves the classification through examples of confounder memes from the subset.

5.2 Qualitative analysis

In Table 8, we demonstrate how RGCL addresses the classification errors associated with confounder memes. Our approach significantly reduces the similarity scores between anchor memes and confounder memes. This shows that 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. By aligning semantically similar memes closer and pushing apart dissimilar ones in the embedding space, RGCL enhances classification accuracy.

6 Conclusion

We introduce Retrieval-Guided Contrastive Learning to enhance any VL encoder in addressing challenges in distinguishing confounding memes. Our approach uses novel auxiliary loss with dynamically 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. Our approach also transfers to different tasks, emphasizing its usefulness across diverse meme domains.

Limitation

Hate speech can be defined by different terminologies, 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 use the definition of hate speech from the two datasets: HatefulMemes (Kiela et al., 2021) and HarMeme (Pramanick et al., 2021a). These datasets adopt Facebook's definition of hate speech [5] to strike a balance between reducing harm and preserving freedom of speech. Tackling the complex issue of how to define hate speech will require a cooperative effort by stakeholders, including governmental policy makers, academic scholars, the United Nations Human Rights Council, and social media companies. We align our research with the ongoing process of defining the hate speech problem and will continue to integrate new datasets, as they become available.

In examining the error cases of our system, we find that the system is unable to recognize subtle facial expressions. This can be improved by using a more powerful vision encoder to enhance image understanding. We leave this to future work.

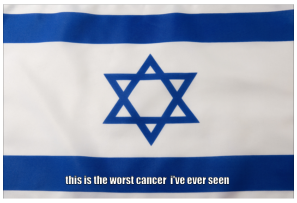
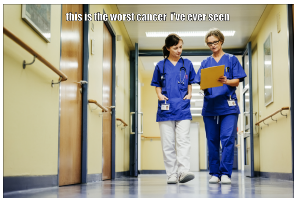
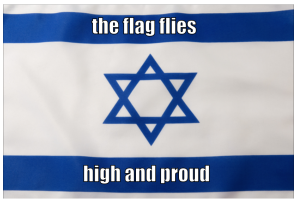
Ethical Statement

Reproducibility. We present the detailed experiment setups and hyperparameter settings in Appendices A and B. The source code will be released upon publication.

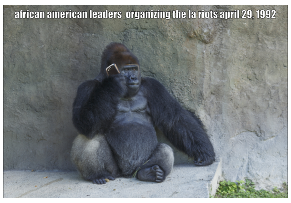
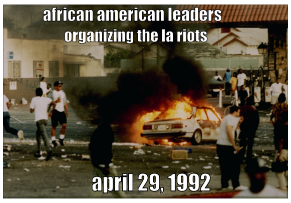
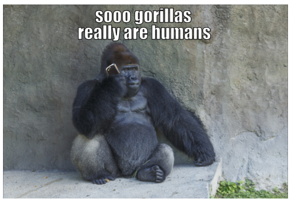
Usage of Datasets. The HatefulMemes, HarMeme, MultiOFF, and Harm-P datasets were

[5] https://transparency.fb.com/en-gb/policies/community-standards/hate-speech/

(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	 african american leaders organizing the la riots april 29 1992	 african american leaders organizing the la riots april 29, 1992	
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)

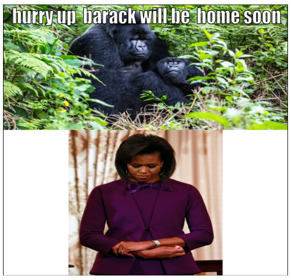
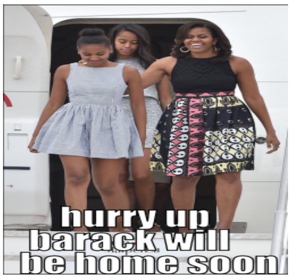
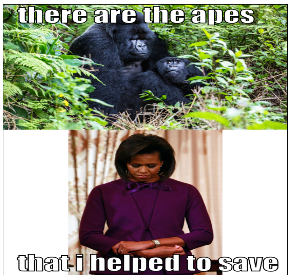
Meme			
HateCLIPper			
Probability	0.385	0.001	0.005
Prediction	Benign ✗	Benign	Benign
Similarity with anchor	-	0.869	0.781
HateCLIPper w/ RGCL (Ours)			
Probability	0.996	0.000	0.000
Prediction	**Hateful** ✓	Benign	Benign
Similarity with anchor	-	**-0.980**	**-0.998**

Table 8: 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.

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.

Societal benefits. Hate speech detection systems like RGCL contribute significantly to reducing online hate speech, promoting safer digital environments, and aiding in protecting human content moderators. These positive impacts, we believe, are substantial and crucial in the broader context of online communication and safety.

Intended use. We intend to enforce strict access controls upon the model release. The model will only be shared with researchers after signing the terms of use. We will clearly state that the system is intended for the detection and prevention of hateful speech. We will specify that it should not be used for any purposes that promote, condone, or encourage hate speech or harmful content.

Implementation consideration. Because our system is based on retrieving examples, multiple retrieval sets reflect different cultural sensitivities that can be applied in reality. Our architecture is well suited to addressing the problem of cultural differences or subjective topics without retraining. However, the annotation of datasets in handling cultural differences or subjective topics needs to be take into consideration before any deployment of systems. The factors need to be considered includes the data curation guidelines, bias of the annotators, and the limited definition of hate speech.

Misuse Potential. Our proposed system does not induce biases. However, training the system on HatefulMemes or HarMeme may cause unintentional biases towards certain individuals, groups, and entities (Pramanick et al., 2021b). To counteract potential unfair moderation stemming from dataset-induced biases, incorporating human moderation is necessary.

Environmental Impact Training large-scale Transformer-based models requires a lot of computations on GPUs/TPUs, which contributes to global warming. However, this is a bit less of an issue for our system, since we only fine-tune small components of vision-language models. Our system can be trained under 30 minutes on a single GPU. The fine-tining takes far less time compared to LMMs. Moreover, as our model is relatively small, the inference cost is much less compared to LMMs.

Acknowledgments

Jingbiao Mei is supported by Cambridge Commonwealth, European and International Trust for the undertaking of the PhD in Engineering at the University of Cambridge.

Jinghong Chen is supported by the Warwick Postgraduate Studentship from Christ's College and the Huawei Hisilicon Studentship for the undertaking of the PhD in Engineering at the University of Cambridge.

Weizhe Lin is supported by a Research Studentship funded by Toyota Motor Europe (RG92562(24020)) for the undertaking of the PhD in Engineering at the University of Cambridge.

Prof. Bill Byrne holds concurrent appointments as a Professor of Information Engineering at Cambridge University and as an Amazon Scholar. This publication describes work performed at Cambridge University and is not associated with Amazon.

We would also like to thank all the reviewers for their knowledgeable reviews.

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.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2009. *Large Scale Online Learning of Image Similarity through Ranking*, volume 5524 of *Lecture Notes in Computer Science*, page 11–14. Springer Berlin Heidelberg, Berlin, Heidelberg.

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].

Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, Tetsuya Sakai, Chen Qu, Hamed Zamani, Liu Yang, W Bruce Croft, and Erik Learned-Miller. 2021. Passage retrieval for outside-knowledge visual question answering. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1753–1757.

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, Yun-hsuan 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].

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. *arXiv*.

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.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

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.

Weizhe Lin, Jinghong Chen, Jingbiao Mei, Alexandru Coca, and Bill Byrne. 2023. Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering. In *Advances in Neural Information Processing Systems*, volume 36, page 22820–22840. Curran Associates, Inc.

Weizhe Lin, Jingbiao Mei, Jinghong Chen, and Bill Byrne. 2024. Preflmr: Scaling up fine-grained late-interaction multi-modal retrievers. (arXiv:2402.08327). ArXiv:2402.08327 [cs].

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.

Thanh-Do Nguyen, Chi Minh Bui, Thi-Hai-Yen Vuong, and Xuan-Hieu Phan. 2023. Passage-based bm25 hard negatives: A simple and effective negative sampling strategy for dense retrieval. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, page 591–599, Hong Kong, China. Association for Computational Linguistics.

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].

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of* *the Association for Computational Linguistics: Human Language Technologies*, page 3715–3734, Seattle, United States. Association for Computational Linguistics.

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*.

Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. Semeval-2020 task 8: Memotion analysis- the visuo-lingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, page 759–773, Barcelona (online). International Committee for Computational Linguistics.

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` [6]. 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. The details for fine-tuniung is covered in Appendix E. All the metrics were reported based on the mean of three runs with different seeds. Due to the limited space in Table 1, we provide more details for our main results here. HateCLIPper with RGCL obtained an accuracy of 78.77 ± 0.25 and an AUC of 86.95 ± 0.21 on HatefulMemes.

B Hyperparameter

The default hyperparameter for all the models are shown in Table 9. The modelling hyperparameter is based on HateCLIPper's setting (Kumar and

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

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 9: 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 HateCLIPper's Architecture

For the i^{th} image and text pair (I_i, T_i), HateCLIPper obtains the feature embeddings f_I and f_T [7] with the pretrained CLIP vision and language encoders. To facilitate the learning of task-specific features, distinct trainable projection layers are employed after the extracted feature vectors to obtain projected features f'_I and f'_T. f'_I and f'_T are vectors of dimension n, which is a hyperparameter to tune. These trainable **projection layers** consist of a feed-forward layer followed by a dropout layer. These feature vectors undergo explicit cross-modal interaction via **Hadamard product** i.e., element-wise multiplication. This fusion process is referred to "**align-fusion**" within the HateCLIPper framework. After the align-fusion, a series of **Pre-Output layers** are employed, comprising multiple feedforward layers incorporating activation functions and dropout layers. These layers are applied to the image and text representation f'_I and f'_T to obtain the final embedding vector $\mathbf{g}_i$. The number of Pre-Output layers is a hyperparameter to tune. We shorthand this process of obtaining the joint embedding vector with $\mathcal{F}(\cdot,\cdot)$ for simplification as denoted in Eq. 1.

D Dataset details and statistics

Table 10 shows the data split for the HatefulMemes and HarMeme datasets. Note that HarMeme is first introduced in Pramanick et al. 2021a, however, in Pramanick et al. 2021b, HarMeme had been renamed to Harm-C. Following the notation of previous works (Cao et al., 2022), we use its original name HarMeme in this paper. The memes in HarMeme are labeled 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.

Table 10: Statistical summary of HatefulMemes and HarMeme datasets

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

In addition to hateful memes classification, we also evaluate the MultiOFF, Harm-P and Memotion7K datasets. Table 11 shows the dataset statistics.

Table 11: Statistical summary of MultiOFF, Harm-P and Memotion7K datasets. Neg. for Negative, Pos. for Positive.

Datasets	Train		Test	
	#Neg.	#Pos.	#Neg.	#Pos.
MultiOFF(Offensive)	258	187	58	91
Harm-P(Harmful)	1534	1486	182	173
Memotion7K				
-Humour	1651	5342	445	1433
-Sarcasm	1544	5449	421	1457
-Offensive	2713	4280	707	1171
-Motivation	4526	2467	1188	690

To access the Facebook HatefulMemes dataset, one must follow the license from Face-

[7] Dropped subscript i for simplicity

book[8].HarMeme and Harm-P is distributed for research purpose only, without a license for commercial use. MultiOFF is licensed under CC-BY-NC. Memotion7K has no specific license mentioned.

E LLaVA experiments

For fine-tuning LLaVA (Liu et al., 2023), we follow the original hyperparameters setting[9] 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.

F 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 example 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 12: 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

G Ablation study on 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 13.

Additionally, another popular loss function for ranking is triplet loss (Chechik et al., 2009; Schroff et al., 2015) which compares a positive example with a negative example for an anchor meme. Our results in Table 13 suggest that using triplet loss performs comparably to the default NLL loss.

Table 13: 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

H 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.

[8] https://hatefulmemeschallenge.com/#download

[9] https://github.com/haotian-liu/LLaVA