Proceedings of the Twentieth International AAAI Conference on Web and Social Media (ICWSM 2026)

Clustering Internet Memes with Metric Learning and Dynamic Modality Weighting

Victoria Sherratt, Nina Dethlefs

Loughborough University
{v.sherratt, n.dethlefs}@lboro.ac.uk

Abstract

This paper presents a two-stage metric learning approach for large-scale clustering of internet memes into pre-defined knowledge categories. We introduce a novel dynamic modality weighting step to adaptively balance the influence of image and text attributes which outperforms other multimodal approaches. We train and evaluate the pipeline across 678,734 memes from KnowYourMeme.com and achieve an F1 score of 91% when assigning unseen memes from a different source to KnowYourMeme.com categories. Our proposed approach incorporates more meme types than prior research, enabling the alignment of individual memes to crucial knowledge sources for information retrieval tasks, with further applications in meme analysis, misinformation detection, hateful meme detection and internet cultural studies.

1 Introduction

Internet memes are multimodal content circulated in online spaces which evolve by users continually editing them. They have been the focus of communications research for some time, with more recent approaches outside of these fields seeking to study internet memes at scale using advances in natural language processing and computer vision (Courtois and Frissen 2023). In computational approaches, research has moved from tasks such as sentiment analysis or classification of internet memes to focus on understanding misinformation spread through internet memes, harmful or hateful content, propaganda, or how they represent cultural and political themes in online spaces (Afridi et al. 2021; Hermida and Santos 2023; Kulkarni 2017).

As memes are context-dependent and referential, external knowledge sources such as KnowYourMeme.com (KYM) have proven beneficial for solving challenging meme-related tasks, and our work is motivated by these recent developments to enrich memes with contextual knowledge (Grasso et al. 2024; Sharma et al. 2023; Sherratt 2022; Tommasini, Illievski, and Wijesiriwardene 2023; Zhu 2020).

By aligning visually and textually diverse memes to existing structured knowledge sources like KYM, memes can be associated with information about its origins, usage conventions, cultural and social significance, and their relationship to other formats or memes. Such added context can aid understanding ambiguous or ironic content in memes and trace ideological or narrative shift across meme variants to support misinformation studies, political communication studies and internet cultural studies. Our method provides a scalable foundation for enriching memes with contextual knowledge by automatically grouping diverse memes into pre-defined categories, even when modalities are noisy or weakly aligned.

Linking memes to existing knowledge sources is challenging due to the visual and textual diversity of memes within defined categories, the volume of memes, and overlapping content. Existing approaches often focus on task-specific clustering that do not align memes to external knowledge sources (Dubey et al. 2018; Theisen et al. 2021; Valensise et al. 2021; Zannettou et al. 2018), or limit retrieval tasks and knowledge integration to visual-only clustering with template-based memes (Joshi, Ilievski, and Luceri 2024).

We instead propose a two-stage supervised metric learning approach using image and text modalities to align memes to existing knowledge sources like KYM to ground ambiguous, referential meme content to established knowledge frameworks and enable retrieval or interpretation. Our contribution includes:

- A scalable metric learning pipeline across over 678,734 memes and 13,552 class labels.
- To the best of our knowledge, we introduce the first clustering method to align internet memes to external knowledge sources that uses both image and text modalities.
- A dynamic modality weighting component to increase accuracy across different test cases and determine modality influence for semantic similarity in internet memes.

We detail related research in Section 2 and data used throughout the paper in Section 3. In Section 4, we outline the two-stage approach and crucial elements of the pipeline. We demonstrate the effectiveness of our method in Section 5 and discuss these results in Section 6.[1]

2 Related Work

Previous work typically clusters memes using either text or image features in isolation, and those that do use mul-

[1]Code: https://github.com/vemchance/memeclustering

Figure 1: Example memes belonging to three KYM archive entry pages with the corresponding KYM class label.

timodal features cluster or classify memes for task-specific goals such as understanding meme popularity, evolution and propagation, or political stance using a sample of memes. Multimodal clustering with the aim to integrate contextual knowledge from external sources like KYM and consider the broader meme landscape, rather than sampled memes for a task-specific analysis, is under-explored.

Text-only clustering approaches include phrase-graph tracking and clustering memes using textual data and metadata (Coscia 2021; Leskovec, Backstrom, and Kleinberg 2009; Simmons, Adamic, and Adar 2021). Ferrara et al. (2013) proposed the term 'protomemes' used in our approach to describe a pre-clustering step based on this textual and meta data, but differs from our pre-clustering step, as our method is based on visual features and is used to facilitate efficient training in the remaining pipeline.

Other studies have developed an automated visual recognition pipeline to map memes of the same genre or used HDBSCAN with a comparable data size of 2 million memes to cluster memes into broad categories from Reddit (Theisen et al. 2021; Valensise et al. 2021). Both approaches, whilst using the visual features of a large number of memes, are an unsupervised clustering of internet memes which does not link them to pre-existing knowledge sources.

Zannettou et al. (2018) utilised pHash to extract image features and a density-based clustering algorithm to cluster memes from various sources. The authors then use images from these clusters to identify which entry page from KYM a meme belongs to and explore how memes evolve and popularity of certain memes in different web communities. In contrast, our research uses the embeddings from both modalities, and is instead optimised for further retrieval tasks as opposed to only exploring propagation and meme evolution.

Dubey et al. (2018) and Beskow, Kumar, and Carley (2020) clustered memes to understand propagation and political ideology respectively, using both textual and visual features. However, both approaches utilise 'image macros', a subset of memes with consistent templates, and whilst they use multimodal embeddings neither method is specifically designed for knowledge retrieval tasks. Guo, Ma, and Zubiaga (2022) similarly leveraged multimodal embeddings to cluster memes for emotion classification, but not for knowledge retrieval tasks. Relating to modality influence, Thakur et al. (2023) explored the influence of image and text modalities in hate speech and misogyny classification for internet memes at a smaller scale; we also demonstrate part of our method can be applied to study modality influence in single memes or defined groups of memes, using a larger sample of memes.

Joshi, Ilievski, and Luceri (2024) clustered memes using a vision transformer to align to an existing knowledge graph proposed by Tommasini, Illievski, and Wijesiriwardene (2023), which also largely focuses on template type memes. In contrast our approach examines a greater number of memes belonging to a KYM entry page; comparatively, this means we examine 13,552 potential classes for memes, in comparison to the 1,375 unique templates described by Joshi, Ilievski, and Luceri (2024).

Additionally, our method incorporates meme text and can cluster memes based on their textual attributes instead of limiting clustering to templates or media frames. With this multimodal approach, our method brings together prior research which has focused on each modality separately by dynamically weighting both modalities to achieve better performance, and covers a broader range of meme types.

3 Data Overview

Memes are digital artifacts that evolve through user participation. Individual memes develop, as described by Shifman (2013) into 'groups of digital objects' that belong together based on their shared characteristics. Archival platforms like KnowYourMeme.com (KYM) group individual meme variants into entries that document their history, context, and cultural relevance.

In Figure 1, we highlight individual meme examples that belong to the KYM entry pages 'Ancient Aliens', 'Does it spark joy?' and 'Dubs Guy'. These individual meme variants, despite visual and textual variations, share thematic or stylistic elements such as recurring characters or repeated phrases. Thus, they developed together and share similar origins and histories, and are grouped together under the one of the three KYM entry pages.

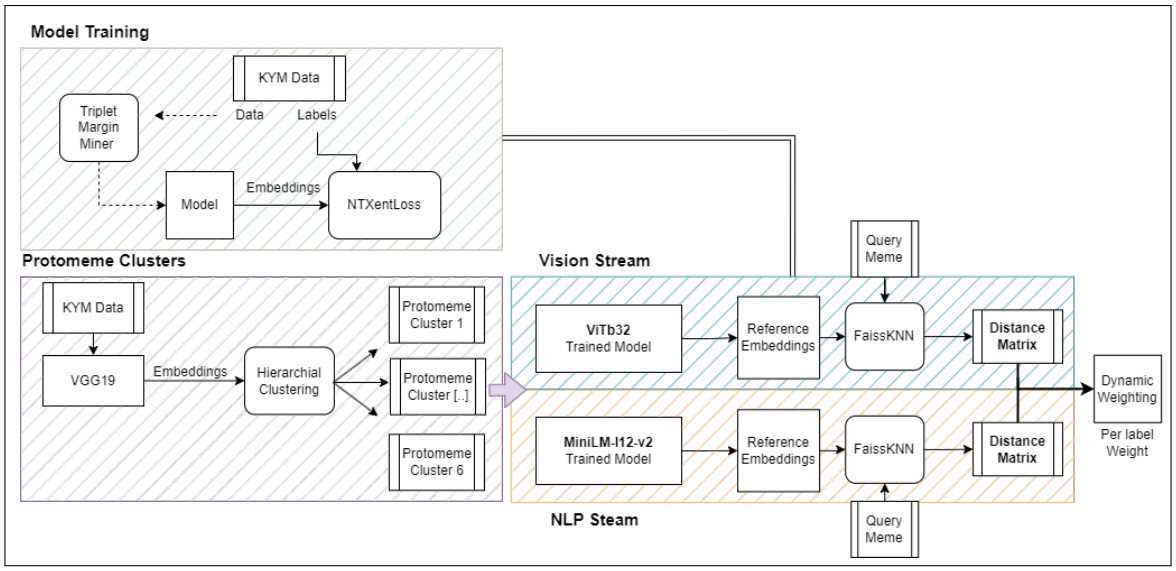

Figure 2: Two-stage metric learning pipeline with pre-clustering (Protomemes), model training and inference.

Throughout, we use the title of the KYM entry page to define the category an individual meme belongs to, and refer to this as the **KYM class**. By aligning individual memes to their respective KYM class, individual memes can be enriched and grounded with contextual information for downstream tasks. However, the intra-class diversity and varying importance of image or text features in meme similarity, as well as the number of potential classes a meme belongs to, makes automatic meme clustering challenging.

3.1 Data Sources

We collected 13,552 KYM entries from the first posted entry up until the final entry as of 27/08/2023 and the individual memes under each entry page, totalling 678,734 individual memes. As in Figure 1, each individual meme is labelled with the entry name it appears under - for example, the first four memes are labelled with the KYM class 'Ancient Aliens'.

KYM entries are categorised as Confirmed (moderated) or Unconfirmed (user-submitted). Whilst Confirmed Memes are approved by moderators, Unconfirmed Memes have yet to be approved; the guidelines for approving memes are not clear, although the KYM website states user-submitted entries are rejected for 'lack of notability or completeness'.[2] There are, however, entries from up to 15 years' ago with little research or example memes,[3] and others that have a large number of examples and detailed research,[4] both of which have yet to be confirmed or rejected. Whilst this dataset may contain KYM classes where variants are disputed as belonging together to form a 'group' of memes, this dataset represents a challenging use-case for our approach.

We exclude entries with fewer than 10 memes, resulting in 344,695 Confirmed memes (4,478 KYM classes) and 334,039 Unconfirmed memes (9,074 KYM classes). We train our models on Confirmed Memes and separately retrain on Unconfirmed Memes to test robustness. For cross-domain testing, we use ImgFlip[5] memes which are labelled with a valid KYM class to evaluate our approach. ImgFlip contains a webpage for meme templates and a gallery of user-generated memes for that template; these ImgFlip webpages have been manually mapped to KYM and this mapping is available in the code base provided. For evaluation, we use 3,966 memes from ImgFlip which belong to 299 KYM classes.

Text Extraction. For all three datasets, we extract meme text using the Google Vision API which is a commonly used OCR extraction method for internet memes.[6] We do not treat the text; prior research has shown that real-world applications of meme research perform poorly 'in the wild' as experimental training data often has cleaner extracted text (Kirk et al. 2021). We use the text data as extracted with no corrections to improve generalisability of our approach outside of training and testing scenarios.

4 Methodology Overview

Our approach consists of two stages: (1) unsupervised pre-clustering based on visual similarity (Protomemes, first described by Ferrara et al. (2013)) and (2) supervised metric learning using both image and text modalities to semantically cluster internet memes to their assigned KYM class. We outline the pipeline briefly in Figure 2 and provide more detail on the Protomeme clusters in Section 4.1, the metric learning approach in Section 4.2 and finally the dynamic modality weighting in section 4.4.

[2]https://knowyourmeme.com/memes/ian-wojak
[3]https://knowyourmeme.com/memes/zeeky-boogy-doog
[4]https://knowyourmeme.com/memes/give-squirtle-a-face
[5]https://imgflip.com/
[6]https://cloud.google.com/vision/

Algorithm 1: Model Training and Embedding Indexing

Input: Protomeme Clusters $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$, KYM Data $\{\mathcal{D}_{\text{image}}, \mathcal{D}_{\text{text}}, \mathcal{D}_{\text{label}}\}$
Output: FAISS Indices $\{\mathcal{F}_{v,i}, \mathcal{F}_{l,i}\}$ for Vision and NLP Streams
for each cluster $\mathcal{C}_i$ in $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ **do**
    Train Vision Stream $\mathcal{M}_{v,i}$ using $\mathcal{D}_{\text{image}}$ and $\mathcal{D}_{\text{label}}$
    Train Text Stream $\mathcal{M}_{l,i}$ using $\mathcal{D}_{\text{text}}$ and $\mathcal{D}_{\text{label}}$
    Generate Vision Reference Embeddings $\mathcal{E}_{v,i} \leftarrow \mathcal{M}_{v,i}(\mathcal{D}_{\text{image}})$
    Generate Text Reference Embeddings $\mathcal{E}_{l,i} \leftarrow \mathcal{M}_{l,i}(\mathcal{D}_{\text{text}})$
    Build Vision FAISS Index $\mathcal{F}_{v,i} \leftarrow \text{FAISS}(\mathcal{E}_{v,i})$
    Build Text FAISS Index $\mathcal{F}_{l,i} \leftarrow \text{FAISS}(\mathcal{E}_{l,i})$
end for
Return: FAISS Indices $\{\mathcal{F}_{v,i}, \mathcal{F}_{l,i}\}$

Algorithm 2: Query Meme KYM Class Assignment

Input: Query Meme q, FAISS Indices $\{\mathcal{F}_{v,i}, \mathcal{F}_{l,i}\}$
Output: KYM Class
for each cluster $\mathcal{C}_i$ in $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ **do**
    Compute distances to embeddings:
    $\mathcal{D}_{v,i} \leftarrow \mathcal{F}_{v,i}(q)$, $\mathcal{D}_{l,i} \leftarrow \mathcal{F}_{l,i}(q)$
    Determine per-class minimum distances:
    $\text{Dist}_v \leftarrow \min(\mathcal{D}_{v,i})$, $\text{Dist}_l \leftarrow \min(\mathcal{D}_{l,i})$
    Combine modalities (DMW):
    Combined Score $\leftarrow \text{DMW}(\text{Dist}_v, \text{Dist}_l)$
end for
Assign the query meme q to KYM class:
Assigned KYM class $\leftarrow \arg\max(\text{Combined Score})$
Return: KYM Class

After the initial pre-clustering (Protomemes) step, we train independent image and text models for each Protomeme cluster. As outlined in Algorithm 1, the VisionTransformer receives the image of the meme and the corresponding label as input, whereas the SentenceTransformer receives the extracted meme text and corresponding label. The trained embeddings are stored in a FAISS (Facebook AI Similarity Search) index as 'reference embeddings.'

Outlined briefly in Algorithm 2, the FAISS index is then used to find the nearest neighbours of a query meme using FAISS K-Nearest Neighbors (Douze et al. 2024; Johnson, Douze, and Jégou 2019). We then calculate the query meme's distance to a all indexed KYM classes for the vision and NLP stream separately. A dynamic modality weighting (DMW) function combines the distances of the vision and text streams and converts these to softmax probabilities. The query meme is then assigned to the KYM class based on the highest probability.

4.1 Protomeme Clusters

We examine whether pre-clustering improves accuracy by grouping visually similar memes before training, inspired by previous successful applications of clustering memes into broad categories (Ferrara et al. 2013; Theisen et al. 2021; Valensise et al. 2021). This initial step groups memes together based on visual similarity such as background colour or overall style or position of panels. For the second step in our metric learning pipeline, these clusters feed each model manageable chunks of data and allows the data miner to efficiently select edge cases that refine the model's discriminative capabilities.

We extract VGG19-based image embeddings and calculate the pairwise distance using cosine similarity between each feature vector (Simonyan and Zisserman 2014). We then use agglomerative clustering (Ward linkage) to create Protomeme clusters with a similarity threshold of 0.8 (Murtagh and Legendre 2014).

As this is an initial pre-clustering step, we ensure a fixed number of clusters (e.g., $k = 6$) by performing a brute-force cut in the dendrogram generated by the agglomerative clustering algorithm. The clustering therefore stops when the hierarchy contains the defined number of clusters, enforced by analysing the tree structure of merged clusters and selecting the optimal k-partition.

	Clusters	Accuracy	% of Samples
ViTb32 Baseline	–	0.682	100%
4 Clusters	Cluster 1	0.847	7.92%
	Cluster 2	0.847	17.93%
	Cluster 3	0.666	21.58%
	Cluster 4	0.631	52.58%
	Average	**0.748**	
5 Clusters	Cluster 1	0.921	7.67%
	Cluster 2	0.849	10.79%
	Cluster 3	0.824	17.35%
	Cluster 4	0.667	20.91%
	Cluster 5	0.631	43.27%
	Average	**0.778**	
6 Clusters	Cluster 1	0.889	7.70%
	Cluster 2	0.916	7.67%
	Cluster 3	0.837	9.65%
	Cluster 4	0.844	10.79%
	Cluster 5	0.669	20.91%
	Cluster 6	0.634	43.27%
	Average	**0.798**	
7 Clusters	Cluster 1	0.874	7.72%
	Cluster 2	0.863	7.75%
	Cluster 3	0.813	9.72%
	Cluster 4	0.818	10.21%
	Cluster 5	0.686	19.00%
	Cluster 6	0.666	21.05%
	Cluster 7	0.627	24.55%
	Average	**0.764**	

Table 1: Accuracy in assigning memes to correct KYM class per Protomeme subsetting strategy.

To determine the optimal number of clusters, we train a ViTb32 model on the memes subset into each Protomeme cluster to assign those memes to the correct KYM class. The model is trained only on the memes within a Protomeme cluster as a specialised learner for the types of memes in each subset. We compare the average performance across each Protomeme cluster (e.g., average performance when subetting into 6 clusters versus 7) to a ViTb32 baseline,

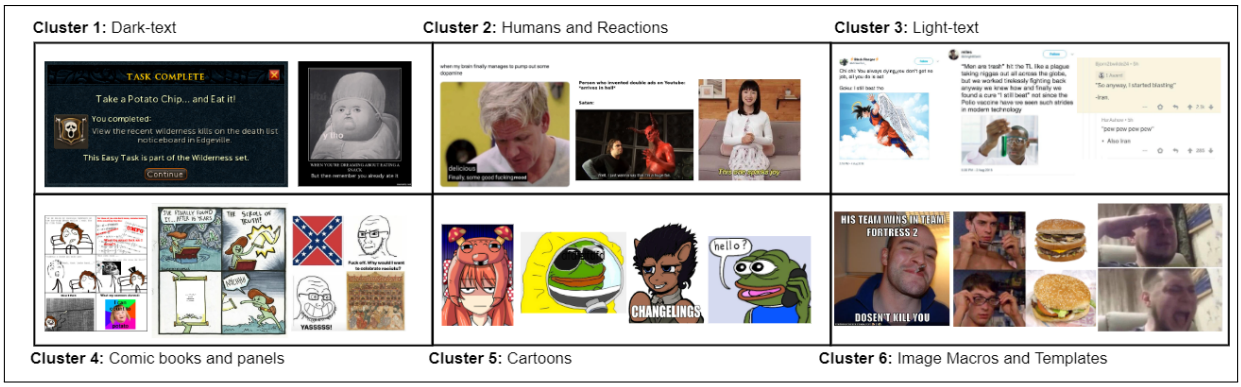

Figure 3: Example memes in Protomeme Clusters.

which is trained on all available data (without subsetting).

We use the same training parameters from the metric learning pipeline briefly outlined in Figure 2. We use 80% of the Confirmed Memes dataset split into a further 80-20 training-test split for this experiment only; 20% of this dataset is reserved in its entirety for the results validation in Section 5.

In Table 1, we report the accuracy of assigning memes to the correct KYM class which have been subset into each Protomeme cluster. We also report the percentage of memes in the Confirmed Memes dataset that are assigned a Protomeme cluster (% of Samples).

Training on the full dataset yields a baseline accuracy of 68.2%, whereas using 6 Protomeme clusters improved average accuracy to 79.8%. Subsetting the total data into 6 Protomeme clusters showed the best balance between cluster coherence and performance gains; accuracy plateaued or declined beyond 7 clusters. In Section 5, we re-validate the effectiveness by comparing models using this pre-clustering method and models trained without pre-clustering.

Protomeme Cluster Descriptions We thematically describe the typical memes in each of the 6 Protomeme clusters as well as a visual representation of sample memes in each Protomeme cluster in Figure 3:

Cluster 1 (Dark-text): Text-heavy memes which can contain a combination of image and text on dark backgrounds, but usually have a more text than typical memes.

Cluster 2 (Humans/Photographic): Memes featuring real-life imagery or humans, often accompanied by large white blocks of overlaid text or subtitles.

Cluster 3 (Light-text): Similar to Cluster 1, but with brighter backgrounds and more dispersed or block-formatted text.

Cluster 4 (Comic Books, Panels): Comic books and panel memes, usually which have a 'hand drawn' aesthetic.

Cluster 5 (Cartoons): Animated memes with minimal text, typically stylised and less reliant on captioning.

Cluster 6 (Macros, Reactions): Broadest category, including common formats such as image macros, snowclones, and reaction images, usually the focus category in prior research.

While KYM classes typically map to a single Protomeme cluster, some entries span multiple clusters. In training, each class is assigned to the cluster containing the majority of its examples. For unseen memes, cluster assignment is determined by computing cosine similarity to cluster centroids.

4.2 Metric Learning Framework

Following Protomeme clustering, we train separate vision and text models on data in each Protomeme cluster using a supervised metric learning approach to optimise semantic similarity within KYM classes. For the vision modality, we use ViTb32 with ImageNet pre-trained weights (Deng et al. 2009). For the text modality, we adopt the SentenceTransformer model MiniLM-L12-v2 with an embedding size of 4096 (Reimers and Gurevych 2019).

Both models are trained independently with the same objective to bring intra-class samples closer in the embedding space and push inter-class samples apart. Experiments with different backbones, such as convolutional neural networks and smaller transformers, are detailed in Appendix A.4 for both text and image modalities.

We use the Pytorch Metric Learning library (PML), which is released under the MIT license, to train the backbone vision and text models independently (Musgrave, Belongie, and Lim 2020). There are two crucial parts of the metric learning training in addition to each pretrained model:

Triplet Margin Miner To efficiently sample informative training examples, we use the Triplet Margin Miner which considers all valid triplets in a batch and focuses the loss on hard negatives within the margin (set to 0.7). This combination consistently produced the best performance in initial trials detailed in Appendix Section A.3.

The weights of the model are updated based on challenging samples, where the negative sample is close to the positive sample despite being from another class, as its similarity falls within the defined margin; in other words, the model is trained on samples that are harder to discriminate.

NTXentLoss We use NTXentLoss (Normalized temperature-scaled cross entropy loss) with cosine as the distance function and a temperature of 0.7 (Sohn 2016). NTXentLoss is a generalisation of NPairLoss which

includes temperature scaling and operates batch-wise, which is more efficient for large datasets. NTXentLoss uses positive pairs (two different samples from the same label) and compares these to negative pairs (samples from a different label). For example, in a batch of size 6 with two classes A and B: $[A_1, A_2, A_3, B_1, B_2, B_3]$, NTXentLoss computes positive pairs and negative pairs as:

Positive Pairs: $(A_1, A_2), (A_2, A_3), (B_1, B_2), (B_2, B_3)$,

Negative Pairs: $(A_1, B_1), (A_1, B_2), \ldots, (A_3, B_3)$.

During training, NTXentLoss ensures that the cosine similarity between the positive pairs $[A_1, A_2, A_3]$ are close in the embedding space while $[B_1, B_2, B_3]$ are separated. We found NTXentloss performed best for our use case, but detail experiments with losses in Appendix A.3.

Training Parameters Training is identical for both vision and text streams and for each Protomeme cluster. We use the AdamW optimizer (learning rate 1e-5, weight decay 0.01) and apply early stopping after 5 epochs without loss improvement. Images are resized to 224×224 pixels. Models are trained on a distributed setup with six NVIDIA RTX 3090 GPUs using PyTorch's DistributedDataParallel. For each cluster, 80% of class samples are used for training and embedding indexing; the remaining 20% is held out for validation in Section 5.

4.3 Dynamic Modality Weighting

To combine predictions from vision and text streams, we introduce a Dynamic Modality Weighting (DMW) mechanism that assigns query-specific weights reflecting how confident each modality is in assigning a meme to the correct KYM class. At such a scale, this approach has not been explored in meme-related research.

After training, embeddings for all reference memes are stored in FAISS indexes for each modality. To assign a query meme to the correct KYM class, we compute distance of the query meme to all indexed memes using K-Nearest Neighbours (KNN) search in both vision and text indexes. For each modality, we determine a query meme's distance to each KYM class by identifying the shortest distance to any indexed meme belonging to that particular class.

These distances are then transformed into modality-specific class membership probabilities using a softmax normalisation. For a given class c, the probability that a query meme belongs to class c according to each individual modality is defined as:

$$p_c = \frac{e^{-d_c}}{\sum_{c'} e^{-d_{c'}}} \tag{1}$$

where d_c is the minimum distance from the query meme to class c in the given modality. The negative exponentiation creates an inverse relationship between distance and confidence so lower distances correspond to higher confidence scores. This provides two sets of confidence scores, from the text modality ($p_{t,1}, p_{t,2}, ..., p_{t,n}$ for n classes) and from the vision modality ($p_{v,1}, p_{v,2}, ..., p_{v,n}$). From these, we identify the highest confidence score from each modality: $p_t = \max(p_{t,1}, p_{t,2}, ..., p_{t,n})$ and $p_v = \max(p_{v,1}, p_{v,2}, ..., p_{v,n})$. These maximum values represent how confident each modality is in predicting which class a query meme belongs to.

We then compute dynamic weights for text (w_t) and vision (w_v) based on these maximum confidence scores:

$$w_t = \frac{p_t}{p_t + p_v + \epsilon} \tag{2}$$

$$w_v = \frac{p_v}{p_t + p_v + \epsilon} \tag{3}$$

Here, ϵ serves as a small constant (10^{-8}) to ensure numerical stability. The combined probability for any class c is then determined by weighting each modality's probability for that specific class:

$$P_c = w_t \cdot p_{t,c} + w_v \cdot p_{v,c} \tag{4}$$

where $p_{t,c}$ represents the text modality's probability that the meme belongs to class c, and $p_{v,c}$ represents the vision modality's probability for the same class. The weights w_t and w_v function as modality-specific coefficients that allocate influence between text and visual features proportional to their respective predictive confidence for the query meme.

Finally, scores are normalized across all classes to produce a probability distribution:

$$\hat{P}_c = \frac{P_c}{\sum_{c'} P_{c'}} \tag{5}$$

At inference time, a query meme is assigned to the class with the highest final probability $\hat{P}_c$.

4.4 Comparison Models

We benchmark our dynamic modality weighting (DMW) approach against individual image and text models and three multimodal architectures: CLIP, FLAVA, and BLIP-2 (Li et al. 2023; Radford et al. 2021; Singh et al. 2022). All models are trained using the same metric learning pipeline for comparability. As FLAVA and BLIP-2 report powerful few and zero shot capabilities, we additionally report the results of both these models *without* fine-tuning to understand the effectiveness of the metric learning pipeline.

CLIP We fine-tune the ViTb32 variant of CLIP to optimise for meme similarity rather than caption-image alignment. We add a trainable scalar parameter α that performs a weighted average of CLIP's image and text embeddings. A lightweight linear transformation and layer normalisation module is added to refine the fused embedding and preserve dimensionality. All pre-trained CLIP layers are frozen except the final projection layer.

FLAVA We utilise the facebook/flava-full pre-trained model and fine-tune this on the task. We preserve FLAVA's native multimodal encoder while adding a linear projection layer that transforms the 768-dimensional multimodal embeddings into a more compact 512-dimensional representation. We maintain FLAVA's original vision and text encoders and the multimodal fusion mechanism, leveraging the pre-trained cross-modal understanding while adapting the output dimension for efficient nearest-neighbour search.

BLIP-2 We use the Salesforce/blip2-opt-2.7b configuration, which combines a vision encoder with a large language model via a querying transformer. We freeze the vision encoder to reduce computational cost and train only the components responsible for multimodal fusion (the query transformer). From the final hidden state, we extract the first token as the condensed representation and apply a projection to 512 dimensions for similarity computation.

5 Results

In this section, we outline the results of training the pipeline on the Confirmed Memes in Section 5.1 and the results of retraining the same pipeline on the Unconfirmed Memes in Section 5.2. Finally, we demonstrate assigning new, unseen memes from a different dataset (ImgFlip) to the existing KYM classes in Section 5.3. Fine-tuned FLAVA and BLIP-2 models are indicated with the notation ^{FT}.

We report the results of assigning memes to correct KYM class per Protomeme clusters. To further evaluate the effectiveness of this subsetting strategy, in Table 2 we report a baseline for the Image and Text models, CLIP and the DMW approach, which performs the task of assigning the entire dataset of Confirmed Memes to the correct KYM class *without* Protomeme subsetting.

	Acc.	P.	R.	F1
CLIP	0.586	0.698	0.711	0.691
Image	0.682	0.679	0.731	0.695
Text	0.607	0.607	0.644	0.618
DMW	0.761	0.715	0.761	0.729

Table 2: Performance of CLIP, ViTb32, MiniLM and DMW on Confirmed Memes dataset (no subsetting).

Performance is evaluated using the macro-averaged precision, recall and F1-score to provide a balanced view of performance across all classes. Accuracy is calculated by using the precision at one metric (first nearest neighbour) for Text, Image, CLIP, FLAVA and BLIP-2, and the highest probability class for the dynamic modality weighting approach.

5.1 Confirmed Internet Memes

On the Confirmed Memes dataset, our DMW approach achieves an average macro F1-score of 84%, outperforming all other approaches on average and within each Protomeme subset. The vision-only model achieves 79%, while the text-only and CLIP models both score around 74%. Fine-tuned BLIP-2 reaches 82% as the closest competitor. Compared to Table 2, subsetting the data into Protomeme clusters on average yields better performance. Additionally, fine-tuning FLAVA and BLIP-2 shows improved performance compared to an untrained model.

Across each cluster, there is varying performance of all models. Memes within Cluster 2 show the highest performance, with all approaches performing well in assigning memes in this group to their KYM class. In Cluster 5, where there is limited text information, CLIP and the text-only model performs poorly.

		Acc.	P.	R.	F1
Cluster 1 Dark-text	Text	0.861	0.899	0.888	0.886
	Image	0.874	0.889	0.889	0.881
	CLIP	0.872	0.867	0.887	0.867
	FLAVA	0.814	0.845	0.866	0.843
	FLAVA^{FT}	0.906	0.914	0.904	0.903
	BLIP-2	0.876	0.907	0.905	0.897
	BLIP-2^{FT}	0.888	0.921	0.907	0.906
	DMW	**0.919**	**0.942**	**0.921**	**0.925**
Cluster 2 Humans, Photography	Text	0.767	0.849	0.817	0.825
	Image	0.906	0.937	0.918	0.923
	CLIP	0.888	0.900	0.902	0.894
	FLAVA	0.864	0.888	0.864	0.870
	FLAVA^{FT}	**0.932**	0.953	*0.930*	0.937
	BLIP-2	0.889	0.925	0.903	0.909
	BLIP-2^{FT}	0.911	0.946	0.915	0.925
	DMW	0.930	**0.963**	*0.930*	**0.942**
Cluster 3 Light-text	Text	0.825	0.893	0.862	0.869
	Image	0.798	0.829	0.838	0.824
	CLIP	0.773	0.811	0.833	0.811
	FLAVA	0.758	0.782	0.831	0.793
	FLAVA^{FT}	0.824	0.837	0.853	0.836
	BLIP-2	0.801	0.826	0.852	0.828
	BLIP-2^{FT}	0.816	0.846	0.859	0.842
	DMW	**0.884**	**0.917**	**0.893**	**0.899**
Cluster 4 Comics, Panels	Text	0.745	0.813	0.784	0.788
	Image	0.831	0.838	0.845	0.834
	CLIP	0.789	0.786	0.803	0.787
	FLAVA	0.687	0.741	0.759	0.742
	FLAVA^{FT}	0.871	0.878	0.861	0.864
	BLIP-2	0.780	0.818	0.818	0.812
	BLIP-2^{FT}	0.801	0.849	0.828	0.833
	DMW	**0.885**	**0.894**	**0.886**	**0.885**
Cluster 5 Cartoons	Text	0.350	0.558	0.466	0.491
	Image	0.651	0.694	0.669	0.667
	CLIP	0.352	0.457	0.494	0.465
	FLAVA	0.472	0.549	0.573	0.547
	FLAVA^{FT}	0.654	0.669	0.666	0.666
	BLIP-2	0.644	0.686	0.688	0.672
	BLIP-2^{FT}	0.673	0.717	**0.707**	0.700
	DMW	**0.687**	**0.756**	0.700	**0.714**
Cluster 6 Macros, Reaction	Text	0.491	0.640	0.595	0.603
	Image	0.513	0.595	0.636	0.604
	CLIP	0.505	0.573	0.630	0.590
	FLAVA	0.492	0.574	0.631	0.588
	FLAVA^{FT}	0.557	0.608	0.649	0.617
	BLIP-2	0.600	0.665	0.702	0.672
	BLIP-2^{FT}	*0.611*	0.681	**0.707**	0.683
	DMW	*0.611*	**0.737**	0.692	**0.700**
Average	Text	0.673	0.775	0.735	0.744
	Image	0.762	0.797	0.799	0.789
	CLIP	0.697	0.732	0.758	0.736
	FLAVA	0.681	0.730	0.754	0.730
	FLAVA^{FT}	0.791	0.810	0.810	0.804
	BLIP-2	0.765	0.804	0.811	0.798
	BLIP-2^{FT}	0.783	0.827	0.820	0.815
	DMW	**0.819**	**0.868**	**0.837**	**0.844**

Table 3: Confirmed Memes Results.

Despite both image and text under performing in Cluster 5, the DMW approach still effectively leverages the per-class strength of each modality to significantly improve performance. For the most challenging and diverse set of memes in Cluster 6, the DMW approach outperforms all other baselines.

5.2 Unconfirmed Internet Memes

We next apply the pipeline to the Unconfirmed Memes, which contain noisier data and many low-sample classes. We evaluate performance on both the full dataset and a filtered version that matches the Confirmed dataset's class-size distribution. The Unconfirmed Memes contains 334,039 across 9,074 classes compared to 344,695 Confirmed Memes across 4,497 classes. The Unconfirmed Memes dataset has a longer 'tail' of classes with fewer samples, notably 2,455 classes with samples below 10 compared to only 777 classes in the Confirmed Memes.

To match the class-size distribution of the Confirmed Memes dataset, we filtered the Unconfirmed Memes by randomly selecting classes from under-represented sample bins (10–19, 20–29, 30–39) in proportion to their Confirmed counterparts. For instance, as the Confirmed set had 585 classes with 10–19 samples, we sampled 585 from the 2,206 Unconfirmed classes in this range. This reduced the Unconfirmed dataset to 4,458 classes and 250,365 memes. The largest reduction occurred in Cluster 6, where 41,006 memes and 4,843 classes were removed. While this filtering improves class balance, it does not necessarily ensure meme quality or completeness; additional examples are discussed in Section A.1.

The distribution of memes roughly remains the same per Protomeme cluster for the Unconfirmed Memes. The most significant changes are in Cluster 4 (Comic book, panels) where the unfiltered Unconfirmed Memes dataset contains only 1.64% of total samples, and the filtered dataset 1.96% of total samples compared to 10.79% in the Confirmed Memes dataset.

In the unfiltered experiment, DMW achieves an F1 of 58.2%, compared to 51.3% for text, 45.7% for vision, and 52.8% for fine-tuned BLIP-2. Although overall performance drops due to noisy labels and sparse classes, DMW maintains a consistent margin above other baselines on average. BLIP-2 shows better performance in Cluster 5; however, on average and within other Protomeme clusters, the DMW approach outperforms.

In the filtered experiment set, DMW reaches 65.9% F1. Clusters with more influence from the text modality (Cluster 1 and 3) benefit significantly for the DMW approach. Cluster 6, however, remains difficult across all models due to high intra-class variability. Notably, fine-tuning slightly reduced performance for FLAVA and BLIP-2 in some clusters. BLIP-2 again performs better in Cluster 5 than the DMW approach.

Despite the caveats of this dataset and degraded label quality, the DMW method achieves good performance and outperforms individual modality models and multimodal baselines. Whilst Cluster 5 and 6 show poor performance for all models, assigning memes to the correct KYM class

		Unfiltered			Filtered		
		P.	R.	F1	P.	R.	F1
Cluster 1 Dark-text	Text	0.649	0.632	0.623	0.720	0.692	0.689
	Image	0.513	0.546	0.514	0.591	0.611	0.585
	CLIP	0.508	0.520	0.495	0.657	0.651	0.636
	FLAVA	0.502	0.522	0.492	0.576	0.597	0.566
	FLAVA^{FT}	0.488	0.518	0.486	0.555	0.579	0.551
	BLIP-2	0.603	0.609	0.585	0.696	0.694	0.675
	BLIP-2^{FT}	0.615	0.617	0.597	0.706	0.690	0.678
	DMW	**0.728**	**0.711**	**0.701**	**0.783**	**0.770**	**0.762**
Cluster 2 Humans	Text	0.589	0.549	0.548	0.668	0.617	0.627
	Image	0.537	0.537	0.520	0.615	0.616	0.602
	CLIP	0.423	0.446	0.416	0.638	0.630	0.617
	FLAVA	0.504	0.508	0.486	0.568	0.570	0.551
	FLAVA^{FT}	0.460	0.485	0.456	0.576	0.583	0.565
	BLIP-2	0.591	0.594	0.572	0.684	0.672	0.658
	BLIP-2^{FT}	0.602	0.601	0.581	0.695	0.681	0.671
	DMW	**0.687**	**0.648**	**0.646**	**0.767**	**0.740**	**0.740**
Cluster 3 Light-text	Text	0.655	0.631	0.623	0.749	0.695	0.704
	Image	0.412	0.441	0.412	0.462	0.483	0.457
	CLIP	0.427	0.446	0.417	0.610	0.600	0.584
	FLAVA	0.424	0.448	0.417	0.505	0.514	0.491
	FLAVA^{FT}	0.403	0.421	0.392	0.554	0.530	0.514
	BLIP-2	0.520	0.535	0.505	0.621	0.617	0.598
	BLIP-2^{FT}	0.528	0.538	0.512	0.626	0.617	0.602
	DMW	**0.679**	**0.663**	**0.651**	**0.760**	**0.731**	**0.730**
Cluster 4 Comics, Panels	Text	0.622	0.627	0.610	0.697	0.698	0.684
	Image	0.593	0.602	0.584	0.681	0.696	0.679
	CLIP	0.591	0.597	0.580	0.650	0.664	0.640
	FLAVA	0.583	0.592	0.569	0.665	0.666	0.653
	FLAVA^{FT}	0.596	0.620	0.594	0.710	0.716	0.695
	BLIP-2	0.667	0.682	0.656	0.770	0.775	0.760
	BLIP-2^{FT}	0.675	0.686	0.666	*0.771*	0.769	0.758
	DMW	**0.711**	**0.709**	**0.695**	*0.771*	**0.779**	**0.765**
Cluster 5 Cartoons	Text	0.431	0.373	0.381	0.465	0.385	0.407
	Image	0.415	0.421	0.406	0.475	0.475	0.463
	CLIP	0.261	0.287	0.263	0.463	0.473	0.455
	FLAVA	0.367	0.382	0.358	0.412	0.431	0.405
	FLAVA^{FT}	0.338	0.357	0.333	0.414	0.428	0.408
	BLIP-2	0.474	0.482	0.459	0.536	**0.549**	0.528
	BLIP-2^{FT}	0.482	**0.497**	**0.473**	0.540	0.540	**0.529**
	DMW	**0.491**	0.463	0.460	**0.546**	0.515	0.514
Cluster 6 Macros	Text	0.319	0.295	0.292	0.393	0.352	0.356
	Image	0.310	0.325	0.305	0.375	0.390	0.372
	CLIP	0.261	0.275	0.255	0.375	0.385	0.367
	FLAVA	0.276	0.292	0.270	0.348	0.360	0.339
	FLAVA^{FT}	0.178	0.204	0.182	0.269	0.296	0.272
	BLIP-2	0.350	**0.363**	0.340	0.438	0.444	0.425
	BLIP-2^{FT}	0.350	0.361	0.340	0.440	0.442	0.424
	DMW	**0.355**	0.356	**0.341**	**0.475**	**0.449**	**0.445**
Average	Text	0.544	0.518	0.513	0.615	0.573	0.578
	Image	0.463	0.479	0.457	0.533	0.545	0.526
	CLIP	0.412	0.429	0.404	0.565	0.567	0.550
	FLAVA	0.442	0.457	0.432	0.513	0.522	0.501
	FLAVA^{FT}	0.410	0.434	0.407	0.513	0.522	0.501
	BLIP-2	0.534	0.544	0.520	0.624	0.625	0.607
	BLIP-2^{FT}	0.542	0.550	0.528	0.630	0.623	0.611
	DMW	**0.609**	**0.592**	**0.582**	**0.684**	**0.664**	**0.659**

Table 4: Unconfirmed Meme Results.

in other subsets ranges from 73-76%. To the best of our knowledge, the Unconfirmed Memes set from KYM is not considered in meme-related research due to the difficulty of modelling with this dataset.

5.3 Matching ImgFlip to KYM

To evaluate cross-domain generalisation, we test our Confirmed Memes-trained models on memes from ImgFlip, a meme generator site where users apply their own text to fixed templates. We assign ImgFlip memes to their corresponding KYM class without retraining the pipeline. ImgFlip memes are popular meme templates, and thus have a verified Confirmed KYM entry; the Protomeme cluster is assigned labeled KYM classes rather than pairwise distance based on a manual mapping of ImgFlip memes to KYM classes.

All memes from ImgFlip are intended to be modified and users are provided with a template. In contrast to previous datasets, ImgFlip memes therefore have a strong and consistent visual style; text is inputted by the user, although it may exist on the template itself.

In Table 5, there is marginal performance loss using the dynamic modality weighting in Cluster 2, 5 and 6; likely, this is because these clusters contain the most 'template' style memes without text present on the image, and therefore the single image modality model performs well.

CLIP significantly underperforms compared to Confirmed Memes results because ImgFlip's user-generated captions often have minimal relationship to visual content, limiting CLIP's image-text alignment approach. However, the DMW method maintains independent modalities and balances their contribution per query, so maintains good performance. While fine-tuned FLAVA and BLIP-2 outperform CLIP, they still underperform compared to the DMW approach or the image-only model.

6 Discussion

Across the three test cases, the dynamic modality weighting often outperforms a single modality and other multimodal baselines, with some exceptions in the ImgFlip use-case and the Unconfirmed Memes dataset. However, in the use-case of assigning unseen memes to KYM classes (ImgFlip), other multimodal architectures underperform compared to the DMW approach. We suggest some potential reasons for this, and considerations for our approach. In Appendix A.2, we also highlight prominent misclassification examples.

6.1 Modality Weighting Benefits

CLIP's pretraining aligns image-text pairs, which is a limitation in meme-related tasks as meme text is not a caption, and often the text is weakly aligned to the image itself or contradictory to produce humour and sarcasm. While FLAVA and BLIP-2 improve on CLIP, both use fixed integration strategies that must serve all examples equally, regardless of whether a particular meme's class membership derives primarily from its visual or textual elements.

In contrast, our DMW approach maintains separate embedding spaces for image and text, and dynamically fuses

		Acc.	P.	R.	F1
Cluster 1 Dark-text	Text	0.618	0.739	0.619	0.635
	Image	0.900	0.982	0.899	0.899
	CLIP	0.532	0.685	0.532	0.580
	FLAVA	0.504	0.577	0.504	0.518
	FLAVA^{FT}	0.849	0.942	0.849	0.862
	BLIP-2	0.676	0.762	0.676	0.711
	BLIP-2^{FT}	0.734	0.859	0.734	0.753
	DMW	**0.921**	**0.980**	**0.921**	**0.941**
Cluster 2 Humans, Photography	Text	0.496	0.651	0.501	0.542
	Image	**0.974**	*0.976*	**0.973**	**0.974**
	CLIP	0.837	0.929	0.837	0.857
	FLAVA	0.661	0.813	0.661	0.671
	FLAVA^{FT}	0.971	*0.976*	0.971	0.973
	BLIP-2	0.663	0.859	0.663	0.693
	BLIP-2^{FT}	0.742	0.869	0.742	0.764
	DMW	0.968	0.974	0.966	0.970
Cluster 3 Light-text	Text	0.645	0.880	0.646	0.721
	Image	0.807	0.837	0.809	0.822
	CLIP	0.337	0.636	0.337	0.408
	FLAVA	0.427	0.634	0.427	0.490
	FLAVA^{FT}	0.674	0.815	0.674	0.704
	BLIP-2	0.551	0.918	0.551	0.646
	BLIP-2^{FT}	0.657	0.948	0.657	0.753
	DMW	**0.940**	**0.990**	**0.938**	**0.952**
Cluster 4 Comics, Panels	Text	0.659	0.800	0.651	0.689
	Image	0.934	0.946	0.931	0.938
	CLIP	0.789	**0.980**	0.789	0.833
	FLAVA	0.681	0.839	0.681	0.709
	FLAVA^{FT}	0.926	0.978	0.926	0.931
	BLIP-2	0.750	0.897	0.750	0.792
	BLIP-2^{FT}	0.791	0.938	0.791	0.841
	DMW	**0.950**	0.962	**0.948**	**0.954**
Cluster 5 Cartoons	Text	0.377	0.587	0.385	0.426
	Image	**0.857**	0.908	**0.854**	**0.867**
	CLIP	0.192	0.400	0.192	0.228
	FLAVA	0.505	0.665	0.505	0.534
	FLAVA^{FT}	0.816	0.899	0.816	0.832
	BLIP-2	0.519	0.811	0.519	0.577
	BLIP-2^{FT}	0.611	0.837	0.611	0.667
	DMW	0.853	**0.930**	0.850	0.858
Cluster 6 Macros, Reaction	Text	0.229	0.392	0.230	0.265
	Image	**0.715**	0.882	**0.719**	**0.763**
	CLIP	0.095	0.243	0.095	0.122
	FLAVA	0.495	0.737	0.495	0.549
	FLAVA^{FT}	0.497	0.743	0.497	0.561
	BLIP-2	0.528	0.755	0.528	0.579
	BLIP-2^{FT}	0.568	0.783	0.568	0.621
	DMW	0.709	**0.890**	0.713	0.759
Average	Text	0.504	0.675	0.505	0.546
	Image	0.864	0.922	0.864	0.877
	CLIP	0.464	0.645	0.464	0.505
	FLAVA	0.545	0.711	0.545	0.578
	FLAVA^{FT}	0.789	0.892	0.789	0.810
	BLIP-2	0.614	0.834	0.614	0.667
	BLIP-2^{FT}	0.684	0.872	0.684	0.733
	DMW	**0.890**	**0.954**	**0.889**	**0.906**

Table 5: ImgFlip Results.

them based on softmax-normalised confidence. This allows the model to down-weight a modality when its features are unreliable, which is advantageous for noisy text (as in ImgFlip) or challenging datasets (as in Unconfirmed Memes).

Modality Influence Alongside improving performance, the DMW enables analysis of modality influence in internet memes. Table 6 highlights image and text contributions per Protomeme cluster and individual classes previously listed in Figure 1. Values represent percentages calculated from aggregated confidence scores, which can be calculated across single samples, classes, Protomeme clusters, or an entire dataset.

Cluster/Class	Text Influence	Image Influence
Dark-text	49.99	50.01
Humans, photography	41.45	58.55
Light-text	47.50	52.50
Comics, panels	37.72	62.28
Cartoons	11.89	88.11
Macros, Reactions	16.98	83.02
Ancient Aliens	16.31	83.69
Does it spark joy?	29.56	70.44
Dubs Guy / Check 'em	12.25	87.75

Table 6: Modality influence across clusters and classes.

While DMW lacks the cross-modal reasoning of unified models, it shows better performance for clustering memes to external knowledge sources and provides insights into modality influence. For tasks requiring cross-modal alignment, CLIP and similar models may remain more suitable.

6.2 K-Nearest Neighbour Truncation

The dynamic modality weighting implementation initially computes distances from a query meme to all reference memes across both image and text modalities, a process with $O(n)$ complexity that may be prohibitively expensive for extremely large datasets. We investigate the effects of neighbour truncation by limiting each modality to its top-k nearest neighbours to understand whether there is trade-off between computational efficiency and performance.

For each modality, we select only the k closest KYM classes by distance. The dynamic weighting mechanism remains unchanged; however, when a class appears in the top-k of only one modality, that modality receives a weight of 1.0 for that respective class. Table 7 shows the the F1 score using 'all' neighbours, 10, 100, 250, and 500 neighbours.

In smaller Confirmed Protomeme clusters, limiting neighbours effectively improves signal-to-noise ratio by filtering out weak signals. For these clusters, the performance difference between using 250 and 500 neighbours is marginal, suggesting diminishing returns beyond a certain neighbour threshold. Conversely, for larger datasets (notably Clusters 5-6), severe neighbour truncation significantly degrades performance. For Unconfirmed (the filtered set), a lower k is better for these large clusters, though performance is still generally limited.

	K@All	K10	K100	K250	K500
Confirmed					
Cluster 1	0.925	0.916	0.913	0.928	0.929
Cluster 2	0.942	0.934	0.921	0.937	0.942
Cluster 3	0.899	0.886	0.851	0.861	0.894
Cluster 4	0.885	0.862	0.836	0.885	0.886
Cluster 5	0.714	0.667	0.579	0.532	0.635
Cluster 6	0.700	0.633	0.578	0.545	0.516
Average	0.844	0.816	0.780	0.781	0.800
Unconfirmed (Filtered)					
Cluster 1	0.762	0.749	0.692	0.692	0.764
Cluster 2	0.740	0.719	0.670	0.667	0.743
Cluster 3	0.730	0.714	0.659	0.686	0.727
Cluster 4	0.765	0.753	0.767	0.777	0.773
Cluster 5	0.514	0.517	0.458	0.424	0.423
Cluster 6	0.445	0.437	0.395	0.384	0.365
Average	0.659	0.648	0.607	0.605	0.632
ImgFlip					
Cluster 1	0.941	0.862	0.857	0.941	0.941
Cluster 2	0.970	0.959	0.965	0.968	0.970
Cluster 3	0.952	0.942	0.867	0.908	0.945
Cluster 4	0.954	0.924	0.890	0.954	0.954
Cluster 5	0.858	0.836	0.799	0.787	0.814
Cluster 6	0.759	0.742	0.647	0.586	0.555
Average	0.906	0.877	0.838	0.857	0.863

Table 7: F1 scores for different k limits across clusters and datasets.

For the ImgFlip use-case, a smaller dataset compared to the Confirmed and Unconfirmed memes, the optimal k seems to closely follow the Confirmed Memes dataset in terms of plateau. To note, for this use-case a query meme from ImgFlip is compared to neighbour classes in the Confirmed Memes dataset, thus k may exceed the number of ImgFlip samples, and still continue to return better performance. We also observe non-monotonic performance patterns:

- **Local Proximity Region** ($k \approx 10$): Captures tight, highly coherent local clusters.
- **Intermediate Boundary Zone** ($k \approx$ 100-250): Incorporates adjacent but potentially less relevant neighbours, often degrading performance.
- **Global Distribution Range** ($k \approx 500$+): Encompasses broader category structure, regaining performance.

Thus, some classes benefit from tight clustering (low k), while others require broader context (high k). Additionally, therefore, some Protomeme subsets (broad groups of memes) benefit from different optimal k setting overall.

Neighbour selection strategies can therefore be tailored to dataset characteristics; while our approach using all neighbours establishes a strong performance baseline, dynamic neighbour selection scaled to dataset size could maintain comparable performance whilst reducing computational complexity. For particularly large-scale applications and future use cases where low-computational costs are critical, there is potential to implement adaptive thresholds as an ex-

tension to our work, as opposed to our approach which prioritises maximising performance.

6.3 Integrating Protomeme Clustering

Our implementation uses 6 Protomeme clusters from VGG19 features with specialised learners per cluster, where inference can be completed by first assigning new memes to the most similar Protomeme cluster before using the appropriate model. This pre-clustering step is ideal for large and diverse meme datasets, but poses some limitations. We consider two approaches for extending our framework:

Soft Clustering with Learnable Protomemes: Rather than using fixed cluster centroids, the initial Protomeme clusters could be learnable parameters in the metric learning pipeline. VGG19 features could be extracted and compared to existing Protomeme clusters and a membership score calculated; whilst this step is similar to the current implementation, full integration would require a meme to be partially assigned to all clusters (e.g., 70% of Cluster 1, 20% of Cluster 3) to allow all specialised models to process a query meme.

Attention-Based Cluster Routing: This could replace explicit clustering with a learnable attention mechanisms that route memes to specialised processing models. For each meme, a backbone vision model would extract initial, broad features and transform these into vectors that can be queried.

Both approaches increase computational complexity but offer potential improvements over the current implementation. However, our dual-stream approach currently outperforms alternatives for this task, and introducing learnable clusters would require both modalities to share gradients; future work for learnable clusters would need to balance the benefits of a separate embedding space with cross-modal gradient flow.

7 Conclusion

We presented a scalable pipeline for meme classification using visual pre-clustering (Protomemes) and supervised metric learning with dynamic modality weighting. Our approach assigns memes to existing target classes, as opposed to unsupervised clustering in prior research, to align memes to external knowledge sources. The DMW approach outperforms single-modality and multimodal baselines across diverse datasets, including noisy and cross-domain settings, and provides insight into modality influence. Fine-tuning FLAVA and BLIP-2 increased performance despite their strong zero-shot capabilities, demonstrating the effectiveness of the second stage metric learning framework.

We also discuss strategic neighbour selection as a computational efficiency mechanism, noting that dataset-specific truncation can maintain performance while reducing processing requirements. Future work could also consider learnable pre-clustering approaches for the Protomemes step, whilst still balancing the benefits of a separate embedding space. While we primarily align memes to KnowYourMeme.com to support future research in enriching memes with contextual information, our approach could be extended to other similarity-based tasks, such as clustering memes based on ideological framing or harmful content, to judge effectiveness and limitations where categories are abstractly defined.

References

Afridi, T.; Alam, A.; Khan, M.; Khan, J.; and Lee, Y. 2021. A Multimodal Memes Classification: A Survey of Open Research Issues. In Ahmed, M.; Santos, D.; Boudhir, A.; Karas, I.; and Sergeyeva, O., eds., *Innovations in Smart Cities Applications*, volume 4, 1451–1466. Karabuk: Springer Science and Business Media Deutschland GmbH.

Beskow, D. M.; Kumar, S.; and Carley, K. M. 2020. The evolution of political memes: Detecting and characterizing internet memes with multi-modal deep learning. *Information Processing & Management*, 57(2): 102170.

Coscia, M. 2021. Competition and Success in the Meme Pool: A Case Study on Quickmeme.com. *Proceedings of the International AAAI Conference on Web and Social Media*, 7(1): 100–109.

Courtois, C.; and Frissen, T. 2023. Computer Vision and Internet Meme Genealogy: An Evaluation of Image Feature Matching as a Technique for Pattern Detection. *Communication Methods and Measures*, 17(1): 17–39.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Douze, M.; Guzhva, A.; Deng, C.; Johnson, J.; Szilvasy, G.; Mazaré, P.-E.; Lomeli, M.; Hosseini, L.; and Jégou, H. 2024. The Faiss library.

Dubey, A.; Moro, E.; Cebrian, M.; and Rahwan, I. 2018. MemeSequencer: Sparse Matching for Embedding Image Macros. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, 1225–1235. Lyon, France: ACM Press. ISBN 978-1-4503-5639-8.

Ferrara, E.; JafariAsbagh, M.; Varol, O.; Qazvinian, V.; Menczer, F.; and Flammini, A. 2013. Clustering Memes in Social Media. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, 548–555. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-2240-9. Event-place: Niagara, Ontario, Canada.

FORCE11. 2020. The FAIR Data principles. https://force11.org/info/the-fair-data-principles/.

Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.

Grasso, B.; La Gatta, V.; Moscato, V.; and Sperlì, G. 2024. KERMIT: Knowledge-EmpoweRed Model In harmful meme deTection. *Information Fusion*, 106: 102269.

Guo, X.; Ma, J.; and Zubiaga, A. 2022. Cluster-based deep ensemble learning for emotion classification in Internet memes. *Journal of Information Science*, 51(1): 265–283.

Hermida, P. C. d. Q.; and Santos, E. M. d. 2023. Detecting hate speech in memes: a review. *Artificial Intelligence Review*, 56(11): 12833–12851.

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

Joshi, S.; Ilievski, F.; and Luceri, L. 2024. Contextualizing Internet Memes Across Social Media Platforms. In *Companion Proceedings of the ACM on Web Conference 2024*, 1831–1840. Singapore Singapore: ACM. ISBN 9798400701726.

Kirk, H.; Jun, Y.; Rauba, P.; Wachtel, G.; Li, R.; Bai, X.; Broestl, N.; Doff-Sotta, M.; Shtedritski, A.; and Asano, Y. M. 2021. Memes in the Wild: Assessing the Generalizability of the Hateful Memes Challenge Dataset. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, 26–35. Online: Association for Computational Linguistics.

Kulkarni, A. 2017. Internet meme and Political Discourse: A study on the impact of internet meme as a tool in communicating political satire. *Journal of Content, Community & Communication Amity School of Communication*, 6.

Leskovec, J.; Backstrom, L.; and Kleinberg, J. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 497–506. Paris France: ACM. ISBN 978-1-60558-495-9.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Murtagh, F.; and Legendre, P. 2014. Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? *Journal of Classification*, 31(3): 274–295.

Musgrave, K.; Belongie, S. J.; and Lim, S.-N. 2020. PyTorch Metric Learning. *ArXiv*, abs/2008.09164.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sharma, S.; Arora, U.; Akhtar, M. S.; Chakraborty, T.; et al. 2023. MEMEX: Detecting Explanatory Evidence for Memes via Knowledge-Enriched Contextualization. *arXiv preprint arXiv:2305.15913*.

Sherratt, V. 2022. Towards Contextually Sensitive Analysis of Memes: Meme Genealogy and Knowledge Base. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 5871–5872. Vienna, Austria: International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3.

Shifman, L. 2013. Memes in a digital world: Reconciling with a conceptual troublemaker. *Journal of computer-mediated communication*, 18(3): 362–377.

Simmons, M.; Adamic, L.; and Adar, E. 2021. Memes Online: Extracted, Subtracted, Injected, and Recollected. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1): 353–360.

Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556.

Singh, A.; Hu, R.; Goswami, V.; Couairon, G.; Galuba, W.; Rohrbach, M.; and Kiela, D. 2022. FLAVA: A Foundational Language And Vision Alignment Model. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15617–15629.

Sohn, K. 2016. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Thakur, A. K.; Ilievski, F.; Ân Sandlin, H.; Sourati, Z.; Luceri, L.; Tommasini, R.; and Mermoud, A. 2023. Multimodal and Explainable Internet Meme Classification. arXiv:2212.05612.

Theisen, W.; Brogan, J.; Thomas, P. B.; Moreira, D.; Phoa, P.; Weninger, T.; and Scheirer, W. 2021. Automatic Discovery of Political Meme Genres with Diverse Appearances. *Proceedings of the International AAAI Conference on Web and Social Media*, 15: 714–726.

Tommasini, R.; Illievski, F.; and Wijesiriwardene, T. 2023. IMKG: The Internet Meme Knowledge Graph. In Pesquita, C.; Jimenez-Ruiz, E.; McCusker, J.; Faria, D.; Dragoni, M.; Dimou, A.; Troncy, R.; and Hertling, S., eds., *The Semantic Web*, volume 13870, 354–371. Cham: Springer Nature Switzerland. ISBN 978-3-031-33454-2 978-3-031-33455-9. Series Title: Lecture Notes in Computer Science.

Valensise, C. M.; Serra, A.; Galeazzi, A.; Etta, G.; Cinelli, M.; and Quattrociocchi, W. 2021. Entropy and complexity unveil the landscape of memes evolution. *Scientific Reports*, 11(1): 20022.

Wang, J.; Song, Y.; Leung, T.; Rosenberg, C.; Wang, J.; Philbin, J.; Chen, B.; and Wu, Y. 2014. Learning Fine-Grained Image Similarity with Deep Ranking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1386–1393. Columbus, OH, USA: IEEE. ISBN 978-1-4799-5118-5.

Zannettou, S.; Caulfield, T.; Blackburn, J.; De Cristofaro, E.; Sirivianos, M.; Stringhini, G.; and Suarez-Tangil, G. 2018. On the Origins of Memes by Means of Fringe Web Communities. In *Proceedings of the Internet Measurement Conference 2018*, 188–202. Boston MA USA: ACM. ISBN 978-1-4503-5619-0.

Zhu, R. 2020. Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution. *arXiv preprint arXiv:2012.08290*.

Paper Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes. Our paper addresses meme clustering and the data is social media images which do not contain personal information that would exacerbate these issues.
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes, the introduction points to relevant sections for claims.
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes, detailed in the Results sections and further detailed in the Discussions section.
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes, we provide a detailed description of the data.
   (e) Did you describe the limitations of your work? Yes, detailed in the Discussion section and concluding remarks.
   (f) Did you discuss any potential negative societal impacts of your work? Yes, we primarily expand on this in the accompanying Ethical Statement below as opposed to the paper itself. As our work is specifically aimed at improving information retrieval for meme-related tasks and internet meme clustering, the specific negative applications of the pipeline have limited societal impact, and discussion of impact potentially from other use-cases not discussed in the paper would detract from the work itself.
   (g) Did you discuss any potential misuse of your work? Yes, again we cover this in the Ethics Statement below which expands on scenarios outside of the scope of the paper for potential misuse.
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? Yes. We provide a sample code base for reproducibility in a public GitHub codebase. Whilst we only provide limited sample data, due to the size of the data used, we do detail the precise method to collect the dataset both in the paper and code base. Since the data collected is public data, it does not contain personal information or meta information at the source it was obtained from. The model architecture is documented in the Methodology sections.
   (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes.
2. Additionally, if your study involves hypotheses testing...
   (a) Did you clearly state the assumptions underlying all theoretical results? NA - No hypotheses testing
   (b) Have you provided justifications for all theoretical results? NA - No hypotheses testing
   (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? NA - No hypotheses testing
   (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? NA - No hypotheses testing
   (e) Did you address potential biases or limitations in your theoretical framework? NA - No hypotheses testing
   (f) Have you related your theoretical results to the existing literature in social science? NA - No hypotheses testing
   (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? NA - No hypotheses testing
3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? NA - No theoretical proofs
   (b) Did you include complete proofs of all theoretical results? NA - No theoretical proofs
4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes, release the GitHub link for the code in the paper. The GitHub public release contains a set of sample data to run experiments. We detail the collection method for data so this can be reproduced, both in the paper and detailed within the code base. Our experiments are trialled across multiple datasets to ensure the proposed methodology is reproducible across different scenarios.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes, detailed in the Methodology section under 'training parameters'.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes, where applicable error bars are reported across multiple experiments in the Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes, the details of the number of GPUs and specification are detailed in the Methodology section under 'training parameters'.
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes, we outline or evaluation metrics and specifications in the Methodology, specifically the training parameters, and the Results section, and use the appropriate metrics to calculate accuracy/F1 for imbalanced classes.
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? Yes, we briefly discuss misclassification and the impacts on information retrieval in the Discussion section, and expand more on this in

the Ethics Statement by considering to impact of retrieving incorrect knowledge for a particular meme.

5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...
   (a) If your work uses existing assets, did you cite the creators? Yes, however we only use pre-trained models and architectures, or Python libraries, and each are appropriately referenced.
   (b) Did you mention the license of the assets? Yes, we stipulate the license for Python libraries used.
   (c) Did you include any new assets in the supplemental material or as a URL? NA - not included in supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? NA - not reusing assets/data in this manner. Data provided on the GitHub is sample data to run the code.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? Yes, we note that the memes in the supplementary materials and in our public codebase may contain offensive content. Due to the size of the dataset and scale of memes used we cannot release the images fully. For the paper itself, any images or text used have been checked to ensure they do not contain offensive content.
   (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see FORCE11 (2020))? NA - not curating new datasets
   (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see Gebru et al. (2021))? NA - not curating new datasets
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...
   (a) Did you include the full text of instructions given to participants and screenshots? NA - crowdsourcing not used
   (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? NA - crowdsourcing not used
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA - crowdsourcing not used
   (d) Did you discuss how data is stored, shared, and deidentified? NA - crowdsourcing not used

Ethics Statement

Our research improves information retrieval for meme-related tasks by clustering internet memes to existing knowledge sources. We discuss potential misuse, the cost of misclassification, and dataset considerations.

Potential Misuse

Our methodology is developed and tested specifically for internet memes, which have characteristics distinct from other image or social media content. The method may work on other datasets, but we have not validated this; results may not generalise outside our tested domain.

Memes often develop as counter-memes or subversions to gatekeep humour within fringe web communities. Bad actors could potentially develop counter-actions to undermine meme analysis, similar to data poisoning attacks in other domains. While this risk is limited—requiring coordinated collective action to change typical meme-sharing behaviour—it remains a potential drawback specific to meme research.

Cost of Misclassifications

Misclassification results in retrieving incorrect information about a meme or assigning it to the wrong cluster, potentially impacting downstream tasks such as harmful content detection. Our method returns distances to all reference memes, allowing researchers to examine borderline cases and better understand relationships between memes based on visual or textual characteristics.

Dataset Ethics

Data Collection Data is compiled from public galleries on KnowYourMeme.com and ImgFlip.com. We collect only images from URLs without retaining metadata; neither website records personal information about submitters. While users may occasionally include personal information in meme text, this is unlikely in our sample. We provide data collection instructions rather than the full dataset.

Offensive Content The full dataset likely contains offensive content given the nature of internet memes. Only a curated sample is released with the code and supplementary materials; the GitHub page notes this caveat. All examples in the paper have been reviewed to ensure they do not contain offensive material.

A Appendix

A.1 Unconfirmed Memes and Confirmed Memes Duplication

As noted in the main body, Confirmed Memes are moderator approved submissions whereas Unconfirmed Memes are user-submissions that have not been approved. Whilst some memes are approved quickly, others that contain substantial research, originality, and many supporting examples are still awaiting approval; in contrast, others with little research and example memes are still Unconfirmed.

Thus, the rules regarding which memes become Confirmed are not clearly defined. Whilst these rules for moderation are likely based on multiple factors, a key difficulty approving some memes seems to be the balance between identifying genuine new groups of memes that have a thematic style, a notable presence in internet culture, and are developed enough not to be considered derivatives of other meme entries.

As a potential use-case, we explore whether our approach can identify duplicate memes in the Unconfirmed entries compared to Confirmed entries. However, this only examines the similarity of Unconfirmed memes to existing categories; many memes can share relationships or close similarity to other memes, but have been approved as a separate entry. Similarly, this approach would not address other issues with the Unconfirmed Memes set, such as poor research, cohesion of memes suggested, or low sample sizes.

	Classes	Image Duplicate	Text Duplicate
Cluster 1	1093	100	294
Cluster 2	911	21	281
Cluster 3	979	17	176
Cluster 4	149	10	49
Cluster 5	1099	56	14
Cluster 6	4843	580	135
Total	**9074**	**784**	**949**

Table 8: Number of Unconfirmed KYM classes with 80% or more samples with a similarity of 0.9 or higher to a Confirmed Meme class.

Before the dynamic weighting is applied, a distance matrix of a query meme to all referenced memes is generated. We compare the Unconfirmed Memes to the Confirmed Memes and use the distance matrix of each modality to identify potential duplicate classes. Duplicates in Table 8 are identified when an Unconfirmed Memes KYM class has 80% or more of its samples with a similarity threshold of 0.9 or above to Confirmed Memes class. This method indicates that 19.10% of the Unconfirmed Memes are similar, based on text and image attributes, to Confirmed Meme classes.

Examining two specific examples, the Unconfirmed Meme class 'Steven Universe "How To" Book'[7] is stylistically similar to the Confirmed 'Peter Parker Reading a Book'.[8] This seems to be a derivative of the 'Peter Parker Reading a Book' meme, but with a strong visual style of its own; rather than a duplicate, it may be considered a 'spin-off'. This is largely based on the image modality similarity, with only a handful of samples from the text modality of this class matching the similarity threshold.

Additionally, based on both textual and image similarity, the Unconfirmed entry 'Unova Confirmed'[9] seems to be related to 'Hoenn Confirmed'.[10] However, despite its similarity, a second entry 'Sinnoh Confirmed'[11] exists and is part of the Confirmed Memes data, suggesting these memes are considered dissimilar enough to warrant its own entry, yet one is awaiting approval and is considered a duplicate from our experiment.

Whilst the criteria for approving a meme seems based on many factors beyond textual or visual similarity, utilising the distance matrices from our approach can facilitate comparing Unconfirmed Memes to existing classes, and aid understanding their relationship to other memes to verify whether they could be considered a derivative of an existing entry or unique enough in origin, development or notability. Such work could also aid moderators of websites like KnowYourMeme.com in improving existing entries by linking more memes together.

[7]https://knowyourmeme.com/memes/steven-universe-how-to-book

[8]https://knowyourmeme.com/memes/peter-parker-reading-a-book

[9]https://knowyourmeme.com/memes/unova-confirmed

[10]https://knowyourmeme.com/memes/hoenn-confirmed

[11]https://knowyourmeme.com/memes/sinnoh-confirmed

A.2 Misclassification Examples

In the top misclassification examples, the text modality is overridden more frequently in dynamic weighting than the image modality. On average, the text modality, despite being correct, has a lower confidence than the image modality. Additionally, there are examples where information from the text is inferred from the image, thus the text modality becomes effective only when information is explicitly stated.

Figure 4: Example 'Sauce?' memes with diverse templates.

For example, the 'Sauce?' class (Figure 4) reports high misclassification rates. The text modality correctly and more confidently predicts this class only when 'sauce' or 'source' is explicitly included; however, in the absence of this, the image modality more confidently assigns these memes to multiple different, incorrect classes. This class may require cross-modal understanding and abstract reasoning to equate 'sauce' with 'source' (users requesting citation) in instances where the phrase 'sauce' is inferred (e.g., its absence in the second meme of Figure 4).

In Figure 5, the top memes highlighted in red from '[Intensifies]' are assigned to 'Descriptive Noise' incorrectly, and vice versa. The text modality correctly identifies certain phrases as a determining factor, but is overall less confident in these predictions, potentially due to other noise such as '[]' appearing in both classes or the word 'ecch' in two samples. The image modality is again more confident, and thus overrides the text prediction. Notably, however, these two classes seem closely related and difficult to distinguish, and would have been better discriminated if the text modality carried higher weighting.

A.3 Pipeline Hyperparameter Selection

In this section, we briefly outline experiments across two crucial hyperparameters used in the paper: the loss function and data miner. Given the number of loss functions and data miners available to test, we conduct these initial experiments using a randomly selected sample of 100 classes from the Confirmed Memes dataset (the 'Trial Set') using VGG19 as the model.

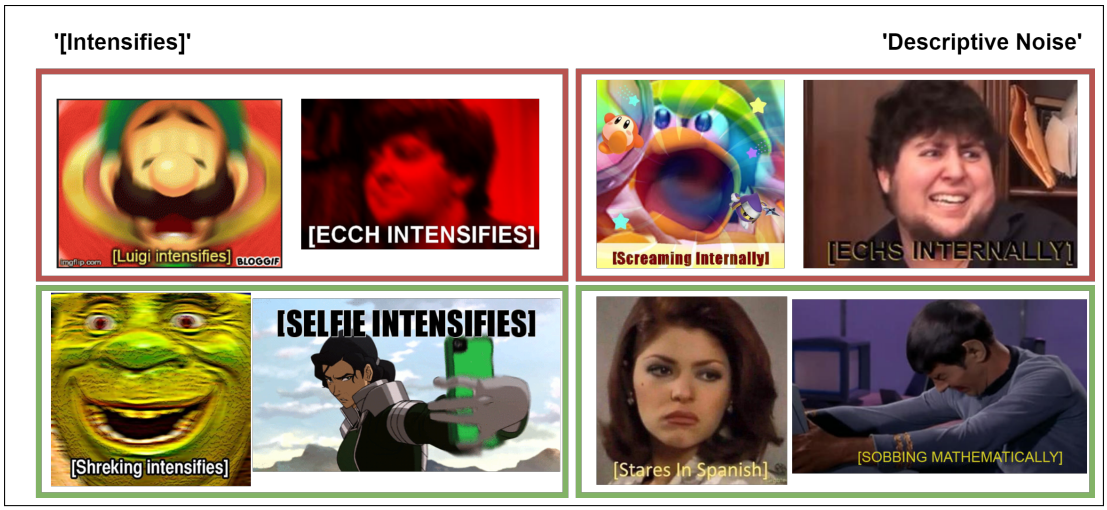

Figure 5: Incorrect (red, top) and correct (green, bottom) KYM class assignments between two commonly confused classes.

Data Miner We test four data miners on a subset of the Confirmed Memes dataset available and documented in the Pytorch Metric Learning library (Musgrave, Belongie, and Lim 2020). For this initial experiment, we use the default settings of each data miner to judge performance. The Triplet Margin Miner outperforms other data miner options in Table 9.

Data Miner	Trial Set Acc.
PairMarginMiner	0.714
UniformHistogramMiner	0.717
TripletMarginMiner	0.725
MultiSimilarityMiner	0.717
Loss Function	
ArcFace	0.747
CosFace	0.748
ContrastiveLoss	0.743
SupCon	0.757
NTXentLoss	0.760
TripletMarginLoss	0.750

Table 9: Miner and loss function accuracy on the Trial Set of Confirmed Memes (100 random classes).

To further tune the data miner, we evaluate the impact of the temperature across all data miner settings (e.g., 'hard', 'semihard' or 'all') against the two best performing loss functions in Figure 6. We find a combination of 'all' triplets with a temperature value of 0.7 generally performs the best.

Loss Function We also test six loss functions again on the Confirmed Memes subset data. For this experiment, Supervised and NTXentloss outperform other loss functions, as detailed in Table 9. For the losses used, Contrastive Loss and Triplet Margin Loss use LpDistance, Supervised Contrastive Loss and NTXentloss use Cosine Similarity.

We also use the default reducer for each loss function, and found modifying the reducer had little impact, which includes the Mean Reducer for NTXentloss and Triplet Margin Loss, and Average Non-Zero Reducer for Contrastive Loss and Supervised Contrastive Loss.

Data Miner and Loss Combinations We retest the highest performing loss functions on the full dataset, in conjunction with the highest performing data miner. Given the Triplet Margin Miner outperforms other data miners, whilst Supervised Contrastive Loss and NTXentLoss have relatively close performance on the subset dataset, we retest each of these loss functions with and without the Triplet Margin Miner only.

Loss Function	Model	Miner	Accuracy
NTXentLoss	ViTb32	No Miner	0.682
	ViTb32	Miner	0.682
	MiniLM-L12	No Miner	0.607
	MiniLM-L12	Miner	0.592
	CLIP	No Miner	0.661
	CLIP	Miner	0.679
SupCon	ViTb32	No Miner	0.635
	ViTb32	Miner	0.638
	MiniLM-L12	No Miner	0.559
	MiniLM-L12	Miner	0.588
	CLIP	No Miner	0.660
	CLIP	Miner	0.672

Table 10: Loss function and Triplet Margin Miner for each modality and CLIP on the full Confirmed Memes dataset.

This experiment is performed without subsetting the data using Protomeme Clustering. We test the loss and miner combinations on each modality model - the vision transformer, the text transformer, and CLIP, in Table 10.

On the full dataset, NTXentLoss outperforms Supervised Contrastive Loss. Generally, the data miner improves performance though there are instances on the full dataset where this is not the case. However, overall, we chose to utilise the

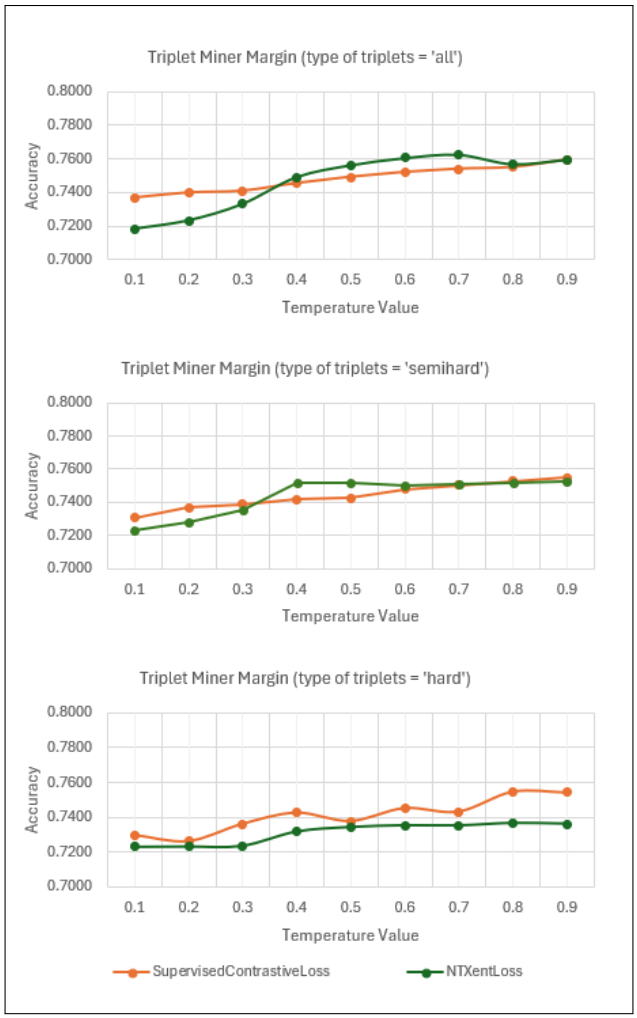

Figure 6: Triplet Margin Miner sampling strategy and margin value performance Trial Set of Confirmed Memes (100 random classes).

data miner as it significantly increases some model performance, e.g., CLIP.

A.4 Vision and Text Backbone

We conducted additional experiments before choosing the vision and text backbone models. In this section, we outline experiments for selecting the vision and text transformers, and their comparison to other models. Experiments are performed on all data within the Confirmed Memes dataset, without Protomeme clustering.

Vision Backbone For the vision modality, we initially tested convolutional neural networks and a three-channel convolutional neural network previously reported to have good success in meme related tasks (referred to as 'DeepRanking') (Beskow, Kumar, and Carley 2020; Wang et al. 2014). We tested all models on a subset of the Confirmed Memes dataset (100 randomly selected classes) and re-tested these on the full dataset.

Whilst we examined variations of VGG, ResNet and ViT (e.g., ResNet50, ViTb16) and various embedding sizes for the vision transformer, we found that VGG19, ResNet101 and ViTb32 achieved the best performance with comparable training time. Larger vision transformer models, such as VitL32, generally performed marginally better as expected but required significant computational overhead for smaller gains.

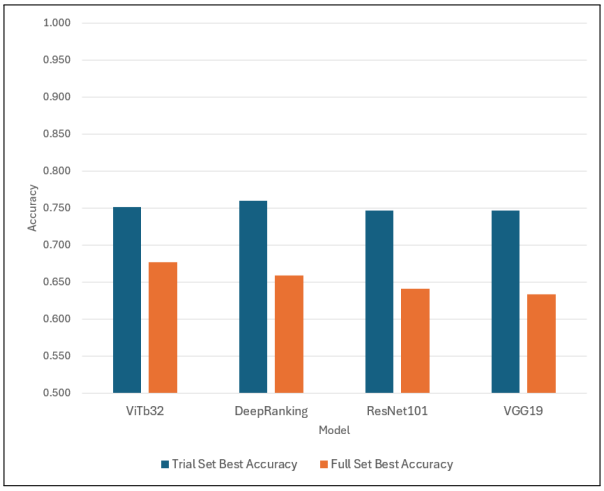

Figure 7: Vision backbone accuracy on trial set and full dataset.

Given the size of the dataset, our focus for selecting a vision backbone was a balance between performance and computational overhead. In Figure 7 we note that the DeepRanking approach performs relatively well on the trial dataset; however, the vision transformer outperforms this model on the full dataset.

Additionally, the structure of DeepRanking was somewhat limiting in our experiments. DeepRanking was significantly slower to train than other models and its structure meant we were limited to using triplet combinations with an offline mining strategy if a data miner was used, rather than exploring other options. Thus, we opted for ViTb32 for both its speed in comparison, better performance, and its adaptability in the pipeline.

Text Backbone For the text modality, we primarily focused on SentenceTransformers which are designed for comparing semantic similarity of text embeddings. We also included BERT as a baseline. We primarily tested the MiniLM base with various embedding sizes, and included the MPNet base - however, we found the MPNet base, despite a higher embedding size, performed as well as the MiniLM in our test case and required significant training time for less performance gain (Reimers and Gurevych 2019).

In Figure 8, we demonstrate the performance of BERT, MiniLM-L6 and MiniLM-L12 with different embedding sizes on the trial set of 100 random classes from the Confirmed Memes dataset, as well as the MPNet base model with an embedding size of 4096 for comparison, over 10

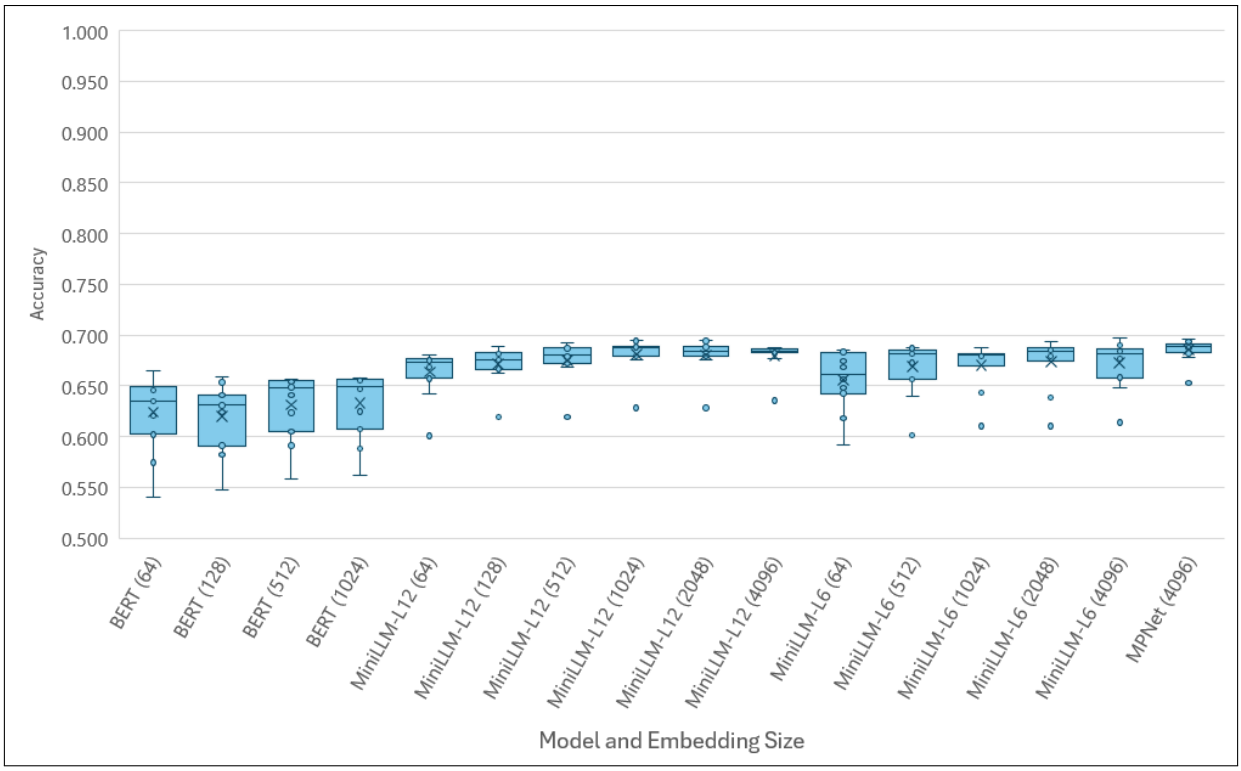

Figure 8: BERT and SentenceTransformer embedding accuracy after 10 epochs on trial dataset of Confirmed Memes.

epochs. We find that the MiniLM-L12 performs best with an embedding size of 2048 or 4096.