Attention-guided Contrastive Hashing for Long-tailed Image Retrieval

Xuan Kou , **Chenghao Xu** , **Xu Yang*** and **Cheng Deng***
Xidian University
{kouxuan98, shydyl1456113691, xuyang.xd, chdeng.xd}@gmail.com

Abstract

Image hashing is to represent an image using a binary code for efficient storage and accurate retrieval. Recently, deep hashing methods have shown great improvements on ideally balanced datasets, however, long-tailed data is more common due to rare samples or data collection costs in the real world. Toward that end, this paper introduces a simple yet effective model named Attention-guided Contrastive Hashing Network (ACHNet) for long-tailed hashing. Specifically, a cross attention feature enhancement module is proposed to predict the importance of features for hashing, alleviating the loss of information originated from data dimension reduction. Moreover, unlike recently sota contrastive methods that focus on instance-level discrimination, we optimize an innovative category-centered contrastive hashing to obtain discriminative results, which is more suitable for long-tailed scenarios. Experiments on two popular benchmarks verify the superiority of the proposed method. Our code is available at: https://github.com/KUXN98/ACHNet.

1 Introduction

Hashing, which can use low-dimensional binary code to represent high-dimensional data, is a crucial embedding technology to improve the efficiency of retrieval and reduce the cost of storage in large-scale image retrieval. Deep hashing methods, such as [Kang *et al.*, 2016b; Yuan *et al.*, 2020], have achieved satisfactory performance in several benchmarks. However, the existing methods are trained under ideal data which is not common in real world, many recent studies [Zhou *et al.*, 2020] demonstrate that we need to deal with unbalanced data.

In a long-tailed dataset, a few categories in the head occupy the vast majority of all samples, while the tail categories have only a very small amount of data. When a network is trained directly on such a dataset, the head categories will dominate, causing the model to ignore the tail categories. In the fields of image classification and object detection, many methods [Ren *et al.*, 2020; Cai *et al.*, 2021] have been proposed to tackle the long-tailed problem. Among them, the performance of re-weighting [Lin *et al.*, 2017], re-sampling [Kang *et al.*, 2016a], and model-based modification methods [Zhou *et al.*, 2020] have obtained promising results. However, these methods are not suitable for long-tailed hashing tasks.

A key problem of long-tailed hashing is the information loss caused by dimension reduction operation, witch will make hash codes less discriminative. In addition, the bias caused by long-tailed data will further affect the retrieval performance. [Chen *et al.*, 2021a] proposed the Dynamic Meta-Embedding (DME) module to solve long-tailed hashing task by transferring knowledge from head to tail categories. However, in DME, only utilizing CE loss will cause follow disadvantages: firstly, the resulting network bias has not been effectively resolved, which may still degrade the ability of the feature extractor. secondly, only completing the classification task does not fully meet the requirement of the retrieval task. More importantly, although the cross-entropy loss based on re-weighting and re-sampling has achieved good results in tasks such as classification [Cai *et al.*, 2021] and object detection [Lin *et al.*, 2017], experiments show that these cross-entropy methods can not improve the discrimination of hash codes [Chen *et al.*, 2021a], this means that we should constrain the hash code to enhance the discrimination. However, the method of adopting contrastive learning to enhance the discrimination of hash codes also faces difficulties on long-tailed data. In a long-tailed distribution, the data in a batch is unbalanced. As shown in Figure 1(b), the lack of tail data leads to the issue of insufficient positive and negative sample pairs, which leads to the deterioration of traditional contrastive learning and is more apparent in the small batch size, so it is not suitable for the long-tailed hashing task.

In this paper, to tackle the challenges mentioned above, we propose an Attention-guided Contrastive Hashing Network (ACHNet), which adopts an attention mechanism and contrastive learning on the hash codes to solve the long-tailed retrieval. The proposed method has two key points: firstly, a Cross Attention Feature Enhancement (CAFE) module, which is established based on the plus-minus values of features, is employed to alleviate the information loss caused by feature dimension reduction and provide a more discriminative hash code for subsequent contrastive learning. Then, a Category-centered Contrastive Hashing (CCH) module is uti-

*Contact Author

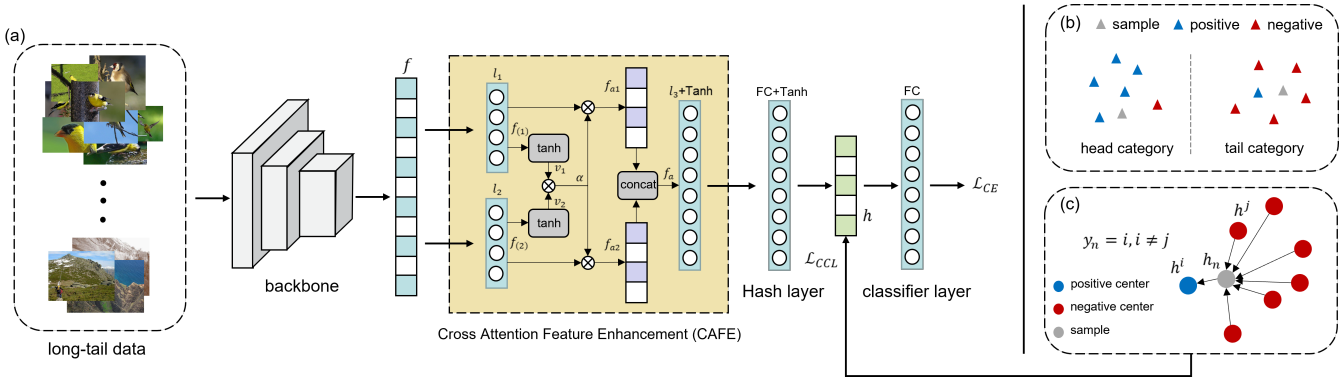

Figure 1: (a) Overview of ACHNet with the cross attention feature enhancement and category-centered contrastive hashing. Firstly, the unbalanced data input the pre-trained feature extractor to obtain feature f, and CAFE is utilized to predict the importance of features for hashing. Then, we utilize the pseudo hash code centers of all categories calculated before the beginning of each epoch for category-centered contrastive learning. Finally, the pseudo hash code is used as the input of the classification layer to complete label prediction. During the test, the pseudo hash code will be converted into binary code through the *sign* activation function. (b) For a long-tailed dataset, the head category and tail category lack negative samples (red) and positive samples (blue) in each batch, so the instance-level contrastive learning is not suitable for unbalanced data. (c) CCH ensures that each image has at least one positive sample and multiple negative samples.

lized to optimize the hash codes to aggregate the hash codes of each category to their center. We conduct comparative experiments on unbalanced Cifar-100 and ImageNet-100 with several benchmarks, under conditions of 32 bits, 64 bits, and 96 bits code lengths, and the results on three different Imbalance Factor (IF) conditions verify the superiority of the proposed method. Our main contributions are as follows:

- We elaborate ACHNet, a network structure including attention module and category-centered contrastive objective, which provides a better hash code for the long-tailed scenario to complete the image retrieval task.
- We propose CAFE to alleviate the information loss originated from dimension reduction and produce more discriminative hash codes. Meanwhile, we exploit CCH to effectively improve the discrimination of hash code in the tail category using the category-centered contrastive loss.
- Our network is compared with multiple deep hashing methods on six benchmarks containing balanced and unbalanced datasets. The results show the superiority of our method in long-tailed image retrieval task.

2 Related Work

Our work is related to long-tailed data learning and hashing. Since the long-tailed hashing is still an opening task, we will introduce the related work from the following two points: long-tailed learning and deep hashing.

Long-tailed learning. In the field of visual recognition, there has been a lot of research on long-tailed learning. We introduce the existing methods from two aspects: the re-balancing methods and the model-based modification methods. In the re-balancing method, the importance of tail category in loss is improved by the re-weighting or re-sampling method. [Chawla *et al.*, 2002] increases the number of tail class samples by over-sampling, [Kang *et al.*, 2016a; Chen *et al.*, 2021b] adopt under-sampling to reduce the number of head class samples. Although the re-sampling method is simple, it is easy to cause overfitting or loss of key samples. [Lin *et al.*, 2017] weights the cross-entropy loss function according to the number of category samples, [Cui *et al.*, 2019] proposes the effective number of samples to weight samples of different categories, and [Cao *et al.*, 2019] reduces the bias of the model to the head by setting a reasonable classification boundary. Then, [Ren *et al.*, 2020] seeks a new paradigm to starts with softmax and proposes meta-softmax method. In the model modification method, the long-tailed learning is carried out by modifying the network or design knowledge transfer module. [Kang *et al.*, 2019] proposes decoupling learning of classifier and feature extractor, [Zhou *et al.*, 2020] designs a double branch network to learn positive distribution data and inverse distribution data respectively. [Wang *et al.*, 2020] classifies samples by learning multiple expert classifiers. More recently, [Cai *et al.*, 2021] is proposed based on [Wang *et al.*, 2020], so that different expert classifiers can learn different categories and complete the training through the one-stage end-to-end method. [Chen *et al.*, 2021a] as the first paper on long-tailed hashing, only cross-entropy loss is adopted and there is no constraint on hash code, although there are many mature related methods. We believe that the long-tailed issue can be better solved by constructing hash layer constraints.

Deep hashing. [Shen *et al.*, 2015] regards deep hashing as a classification problem and obtains hash codes by training classifiers. [Gui *et al.*, 2017] improves it by enhancing the robustness of the network and [Li *et al.*, 2018] attempts to improve performance by using deep networks. In addition, there is another kind of method that regards hashing as a regression problem, such as [Liu *et al.*, 2012]. In these methods, [Kang *et al.*, 2016b] designs a unique fast discrete optimization algorithm, [Yuan *et al.*, 2020] brings the hash code closer to the preset center, but ignores the correlation between categories, which has a particularly significant im-

pact on the case of long-tailed data distribution. There are other kinds of methods, [Wang *et al.*, 2015] directly optimizes nDCG metrics to obtain higher quality hash codes and [Deng *et al.*, 2018] learn hashing functions through triples.

3 The Proposed Method

In this section, we introduce the framework of ACHNet. We first formulate the long-tailed hashing task (Section 3.1), then we elaborate on the cross attention feature enhancement module (Section 3.2) and category-centered contrastive hashing (Section 3.3). Finally, we describe the total loss function and the training algorithm of ACHNet (Section 3.4).

3.1 Problem Formulation

Give a set of images $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}_{n=1}^{N}$ and corresponding labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}_{n=1}^{N}$ from C different categories, deep hashing task wants to learn a feature extractor $\mathcal{E}$ parameterized by θ and a hashing function $\mathcal{H}$, where N is the number of all image-label pairs. First, we can obtain the features of the images $f = \mathcal{E}(x_n \mid \theta), f \in \mathbb{R}^{d_f}$ through the feature extractor. Then, we adopt hashing function $\mathcal{H}$ to obtain the hash code $b = \mathcal{H}(\boldsymbol{f})$ of the corresponding feature, where $b \in \{\pm 1\}^q$, q refers to the hash code length.

In the case of long-tailed learning, the sample number of each category is different. Head categories account for the vast majority of all samples, and the remaining tail categories contain only a small amount of data. The imbalance factor is used to indicate the imbalance degree of long-tailed data. We assume that $\mathcal{S} = \{s_1, s_2, \ldots, s_c\}$ represents the number of images in each category, where s_1 is the largest number of samples in head category, the remaining categories decreased in turn. We follow the imbalance setting and adopt *Zipf's law* to set the number of samples [Chen *et al.*, 2021a].

3.2 Cross Attention Feature Enhancement

In long-tailed hashing, two crucial issues are information loss caused by dimension reduction and head category bias caused by unbalanced distribution. In this paper, we propose two modules to solve these problems. Firstly, CAFE is used to prevent information loss and provide more discriminative hash codes.

In deep hashing, we want to utilize a low dimensional binary code b to represent an image, however, the dimensional difference between a feature and its binary code determines that this mapping is a dimension reduction operation: $b \in \mathbb{R}^q \leftarrow f \in \mathbb{R}^{d_f}$. When high-dimensional data becomes low-dimensional data, some key information will be lost, which will make the hash codes less discriminative. To alleviate this problem, we design a cross attention feature enhancement module to predict the importance of features and provide a more discriminative hash code for the next category-centered contrastive hashing.

As shown in Figure 1(a), our module consists of three full connection layers l_1, l_2, and l_3, and their weight matrices are $W_1 \in \mathbb{R}^{d_f \times d_{f_{(1)}}}$, $W_2 \in \mathbb{R}^{d_f \times d_{f_{(2)}}}$ and $W_3 \in \mathbb{R}^{d_{f_a} \times d_{f_{(h)}}}$, where $d_{f_{(1)}}$ and $d_{f_{(2)}}$ represent the output dimensions of l_1 and l_2, $(d_{f_{(1)}} = d_{f_{(2)}})$, d_{f_a} denotes the dimension of attention feature after catenate operation, and $d_{f_{(h)}}$ denotes the output dimension of l_3. We first obtain the output $f_{(1)}$, $f_{(2)}$ as intermediate features, which is formulated as::

$$f_{(1)} = fW_1, f_{(2)} = fW_2. \tag{1}$$

Considering that the hash code is binary, i.e., $\{+1, -1\}$, we believe that constructing the attention module based on the plus-minus sign of features can effectively help the same category to obtain more similar hash codes. Therefore, in our module, the two full connection layers l_1 and l_2 share the same initial parameters, which will ensure that the initial intermediate features have the same sign, then we adopt the operation shown in Figure 1(a) to perform the attention mechanism. Specifically, we use the *tanh* activation function to obtain the sign vectors v_{s_1} and v_{s_2} representing plus-minus sign of $f_{(1)}$ and $f_{(2)}$. There are two reasons why we don't adopt the *sign* function here: Firstly, it will affect the backpropagation of the network. In face, we want to make the intermediate features interact with each other, rather than just focusing on the key areas. Secondly, the *sign* function can only produce discontinuous binarization results. In contrast, the output of *tanh* exists continuously between ± 1. The attention vector α is obtained by multiplying sign vectors as:

$$\alpha = v_{s_1} v_{s_2}. \tag{2}$$

Then, α is multiplied by the intermediate feature to obtain new feature vectors. Finally, the new features are concatenated and as input to the full connection layer l_3 to generate the feature f_h after applying the attention mechanism:

$$f_a = \mathrm{Concat}\left(\alpha f_{(1)}, \alpha f_{(2)}\right) W_3. \tag{3}$$

where $\mathrm{Concat}(\cdot)$ refers to concatenate. Due to the above operation, the outputs of the two fully connected layers are very similar during training, which will make the plus part in α represent the elements with the same sign in $f_{(1)}, f_{(2)}$, while the minus part is just the opposite, and the value is close to 0. Therefore, f_a will pay more attention to the part that determines the plus-minus sign of hash codes, which will also provide a more discriminative center for the next stage of contrastive learning.

3.3 Category-centered Contrastive Hashing

In deep hashing, the existing contrastive learning methods are not suitable for long-tailed data. When training on unbalanced datasets, the scarcity of tail data will lead to the lack of positive and negative sample pairs, which will cause the instance-level contrastive loss, such as [Chen *et al.*, 2020] not applicable to long-tailed hashing. Meanwhile, the category-level contrastive loss adopted in [Yuan *et al.*, 2020] utilizes the preset hash codebook and ignores the correlation between categories, which is particularly important for long-tail data. Therefore, to solve the above issues, we propose category-centered contrastive hashing.

Specifically, we adopt pseudo hash code center instead of the preset hash codebook to solve maladjustment of contrastive learning on long-tailed data. The process to obtain the pseudo hash center can be formulated as:

$$h^i = \frac{1}{s_i} \sum_{n=1}^{s_i} h_n, \tag{4}$$

where i represents the category, and h^i represents the hash code center of the ith category. Thus, we can obtain the hash code centers of all c categories $\left\{h^1, h^2, \ldots, h^c\right\}$. Compared with the preset hash codebook, correlation between categories is preserved, and the hash codes of tail categories can be aggregated better. As shown in Figure 1(c), when we exploit hash code center as the positive and negative sample-pair, it is guaranteed that each sample has a positive sample and several negative samples.

Because Hamming distance is calculated by binary codes, so it is non-differentiable, the distance between pseudo hash codes h cannot be measured for training. Hamming distance is calculated as follow:

$$D\left(b_i, b_j\right) = \frac{1}{2}\left(q - b_i^\top b_j\right). \tag{5}$$

Analyzing Eq. (5) geometrically, we can find that the relationship between dot product $b_i^\top b_j$ and cosine similarity of b_i, $b_j$can be interpreted as:

$$\cos\theta_{ij} = \frac{b_i^\top b_j}{\|b_i\| \, \|b_j\|}, \tag{6}$$

cosine similarity is differentiable and inversely proportional to Hamming distance, so we adopt cosine similarity to measure the similarity between hash codes. The category-centered contrastive loss of nth image is calculated as follows:

$$L_{CCL} = -\log \frac{\exp\left(d\left\langle h_n, h^i\right\rangle / \tau\right)}{\sum_{j \in 1, C, j \neq i} \exp\left(d\left\langle h_n, h^j\right\rangle / \tau\right)}, \tag{7}$$

where $d(\cdot, \cdot)$ is the cosine similarity distance, τ is the temperature parameter. The hash code will be close to the hash code center, and enough positive and negative sample pairs can also help to distinguish the tail categories.

3.4 Model Training

At the end of ACHNet, We set a classifier layer behind the hash layer to complete classification task. This setting is based on the idea that better hash codes should also have better classification results, the classifier layer will help to obtain a better hash code. The total loss function is shown as follow:

$$L = \beta L_{CE}(\hat{\mathbf{y}}, \mathbf{y}) + L_{CCL}, \tag{8}$$

where $\hat{\mathbf{y}}$ represents the prediction label of the classifier, and β is a hyperparameter. The training algorithm of our method is shown in Alg.1, it represents a computation flow in an epoch.

Algorithm 1 Training algorithm of ACHNet

Input: Training set D, pre-trained feature extractor $\mathcal{E}(\cdot \mid \theta)$, epoch number T, hyperparameter β, W_1, W_2, W_3 belong to attention module $\mathcal{G}(\cdot \mid \theta_m)$ and hash layer $(\cdot \mid \theta_h)$, classifier $(\cdot \mid \theta_c)$

Output: The proposed ACHNet

for $n = 1$ **to** N **do**
  $f_a = \mathcal{G}(\mathcal{E}(x_n \mid \theta) \mid \theta_m)$
  $h_n = (f_a \mid \theta_h)$
end for
Compute the center of each category $\left\{h^1, h^2, \ldots, h^c\right\}$ via Eq. (4)
for epoch minibatch $\mathcal{X}$ **do**
  $f_{(1)} = W_1\mathcal{X}, f_{(2)} = W_2\mathcal{X}$
  $\alpha = tanh(f_{(1)}) * tanh(f_{(2)})$
  $f_a = W_3 Concat(\alpha * f_{(1)}, \alpha * f_{(2)})$
  $h_{\mathcal{X}} = (f_a \mid \theta_h)$
  $\hat{\mathbf{y}} = SoftMax((h_{\mathcal{X}} \mid \theta_c))$
  Compute L_{CE} and L_{CCL}
  $L = \beta L_{CE}(\hat{\mathbf{y}}, \mathbf{y}) + L_{CCL}$
  Update network parameters through optimizer
end for
return ACHNet

		N_{base}	N_{query}	N_{train}
Cifar-100	IF1	50K	10K	50000
	IF50	50K	10K	3732
	IF100	50K	10K	2598
ImageNet-100	IF1	13K	5K	10000
	IF50	13K	5K	9437
	IF100	13K	5K	6834

Table 1: Details of Dataset

4 Experiment

We conduct extensive experiments to verify the superiority of ACHNet on both balanced and long-tailed benchmarks under 32 bits, 64 bits, and 96 bits code lengths.

4.1 Datasets

Cifar-100. Cifar-100 is a dataset widely used in the field of image classification and retrieval. It contains 60000 images at 32×32 from 10 categories. We use 50000 images as the database, with an average of 500 images per category, and the remaining 10000 images as the query set, with 100 images per category. Then we follow *Zipf's law* to randomly sample the training set from database, building three unbalanced benchmarks according to IF=1, IF=50, and IF=100.

ImageNet-100. ImageNet-100 is one of the most popular datasets, it contains more than 120000 images in 1000 categories. According to the setting in [Chen *et al.*, 2021a], we select 100 categories from all 1000 categories to build an unbalanced data set. The database has 13000 images in total and 1300 images in each category, while query set has 5000 images in total and 50 images in each category , the training set is randomly selected from the database. It is worth mentioning that when IF=1, for fair comparison, we only take 100 images of each category as the training set. The details of the datasets are shown in Table 1.

4.2 Implementation Details

We evaluate the proposed method on six benchmarks. Due to the research on long-tailed hashing still being an opening task, in addition to the existing long-tailed hashing methods, we also compare ACHNet with four deep hashing methods. They are DPSH [Li *et al.*, 2016], HashNet [Cao *et al.*, 2017], DSDH [Li *et al.*, 2017], CSQ [Yuan *et al.*, 2020] and LTHNet [Chen *et al.*, 2021a]. Reference to the previous methods, we adopt mean average precision (mAP) as the evaluation metric for all experiments.

IF	IF1			IF50			IF100		
Hash bits	32 bits	64 bits	96 bits	32 bits	64 bits	96 bits	32 bits	64 bits	96 bits
DPSH	0.3113	0.4506	0.4957	0.1069	0.1407	0.1634	0.0978	0.1216	0.1383
HashNet	0.4380	0.5719	0.6311	0.1726	0.1950	0.2079	0.1444	0.1559	0.1631
DSDH	0.5398	0.6100	0.6407	0.1119	0.1000	0.0999	0.0940	0.0872	0.0807
CSQ	0.7711	0.7984	0.7821	0.2221	0.2745	0.2669	0.1716	0.1992	0.1658
LTHNet(k=0)	0.8195	0.8336	0.8400	0.2427	0.3028	0.3309	0.1752	0.2240	0.2415
LTHNet(k=3)	**0.8268**	**0.8416**	**0.8490**	0.2687	0.3354	0.3484	0.1819	0.2376	0.2620
ACHNet	0.8218	0.8299	0.8314	**0.3075**	**0.3624**	**0.3708**	**0.2246**	**0.2770**	**0.2957**

Table 2: The retrieval performance of all comparison methods on Cifar-100 under different IF settings and code length

IF	IF1			IF50			IF100		
Hash bits	32 bits	64 bits	96 bits	32 bits	64 bits	96 bits	32 bits	64 bits	96 bits
DPSH	0.4887	0.6055	0.6514	0.2186	0.3125	0.3791	0.1788	0.2832	0.3468
HashNet	0.4410	0.6006	0.6421	0.3465	0.4034	0.4240	0.3101	0.3770	0.3800
DSDH	0.6554	0.7015	0.7231	0.2568	0.2617	0.2744	0.1841	0.2134	0.2429
CSQ	0.8507	**0.8733**	0.8657	0.6629	0.7022	0.6823	0.5989	0.5620	0.5495
LTHNet(k=0)	0.7924	0.8267	0.8382	0.7369	0.7804	0.7920	0.6771	0.7350	0.7528
LTHNet(k=3)	0.8142	0.8453	0.8592	0.7612	0.8007	0.8157	0.7146	0.7665	0.7828
ACHNet	**0.8592**	0.8702	**0.8779**	**0.8265**	**0.8427**	**0.8472**	**0.7965**	**0.8128**	**0.8163**

Table 3: The retrieval performance of all comparison methods on ImgeNet-100 under different IF settings and code length

We utilize resnet34, which has been pre-trained on ImageNet, as the backbone of all methods. To be fair, we adopt the same hyperparametric setting as [Chen *et al.*, 2021a]. RMSprop algorithm is used to optimize model parameters. On the Cifar-100, we set the learning rate as 1e-5. On the ImageNet, we set the learning rate of the feature extractor as 1e-6 and the rest as 1e-4, the weight decay is 5e-4. we adopt the cosine annealing strategy to adjust the learning rate within each epoch, the final learning rate will be 0.01 times the initial value. Other hyperparametric are set as follows: batch size=8, β=0.2 for Cifar-100 and β=0.8 for ImageNet-100, the total number of epochs is 100.

4.3 Results and Analysis

The experimental results on the six benchmarks are shown in Table 2 and Table 3. It can be seen that ACHNet has been greatly improved on most benchmarks, which verifies the superiority of ACHNet. On Cifar-100, the retrieval performance of our method is particularly significant on unbalanced benchmarks. Under the setting of IF=100, The method achieves 3.3% to 4.3% gains on different code lengths, while under the setting of IF=50, the improvement effect is also between 2.2% and 2.9%. But on the balanced dataset, our method is slightly worse than [Chen *et al.*, 2021a], it shows the superiority of our method on long-tailed distribution data. On the ImageNet-100 dataset, our method surpasses [Chen *et al.*, 2021a] on three training sets with different IF settings. By analyzing the experimental results, we can find that the results of ACHNet can always obtain the maximum improvement in the case of small code length, which verifies the effectiveness of the attention module in curbing dimension reduction operation loss information. In a conclusion, compare with LTHNet on unbalanced data, our method is more suitable for the long-tailed hashing task.

4.4 Ablation Study

As mentioned above, ACHNet has two key points, the cross attention feature enhancement, and the category-centered contrastive hashing. We verify the effectiveness of the proposed method through three experimental settings as follow:

Cross attention. The cross attention feature enhancement module relies on the cross attention mechanism we conceived, which predict the importance of feature based on plus-minus sign of the intermediate features. In order to verify the effectiveness of this operation, we compare CAFE with the Two Branch Module (TBM) composed of only three linear layers, CCH is used in both networks. The experiment compares the retrieval performance of the two modules on ImageNet-100 and Cifar-100 datasets and sets IF=100. The results are shown in Table 4. By analyzing the experiment results, we can see that on the two unbalanced datasets, the cross attention mechanism we proposed effectively improves the performance level of the network without adding any parameters, and the effect is more significant on ImageNet- 100, which confirms the effectiveness of CAFE.

Hash code center. Different from the previous methods, we believe that the operation using pseudo Hash Code Center (HCC) is more effective than the Preset Hash Code Center (PHCC) under long-tailed data distribution. Therefore, we

Hash bits		32 bits	64 bits	96 bits
Cifar-100	TBM	0.2209	0.2759	0.2901
	CAFE	**0.2246**	**0.2770**	**0.2957**
ImageNet-100	TBM	0.7852	0.7974	0.7958
	CAFE	**0.7965**	**0.8128**	**0.8163**

Table 4: Retrieval performance to verify the effectiveness of cross attention mechanism

Hash bits		32 bits	64 bits
Cifar-100	PHCC	0.1837	0.2489
	HCC	**0.2246**	**0.2770**
ImageNet-100	PHCC	0.7848	0.8104
	HCC	**0.7965**	**0.8128**

Table 5: Retrieval performance to verify the effectiveness of hash code center

use the pseudo hash code and the preset hash code in [Yuan *et al.*, 2020] to experiment on our long-tailed hashing network to verify the correctness of our theory. Setting IF=100 and the retrieval performance is shown in Table 5. By analyzing the results, we can find that using our pseudo hash code center can get better results, with an average improvement of 3.5% and 0.7% on Cifar-100 and ImageNet-100, respectively.

Module verification. To verify the improvement of the model by CAFE and CCH, the following four groups of comparison experiments are set: (1) Only Cross-entropy Loss (CELoss) is used without CAFE and CCH; (2) CELoss and CAFE are used without CCH; (3) CELoss and CCH are used without CAFE; (4) Attention-guided Contrastive Hashing network we proposed. The experimental results are shown in Table 6. We can find that both CAFE and CCH improve the retrieval results of the network. The results verify the effectiveness of our proposed CAFE and CCH.

4.5 Parameter Sensitivity

We observe the effect of hyperparameter β on retrieval performance through experiments. On the Cifar-100 and ImageNet-100 datasets, we compare the impact of different hyperparameter settings on ACHNet under different hash code lengths and set IF=100. The experimental results are shown in Figure 2. We can see that due to the different image quality and the minimum number of samples in the tail category of the two datasets, the impact of beta is also different. On Cifar-100, when we set $beta = 0.2$, the result is the best, and much better than other β values setting. This proves that category-centered contrastive loss plays a more important role than CEloss on Cifar-100. In addition, by analyzing the results of the three code lengths, we find that when the code length is the largest, the information loss caused by the dimensionality reduction operation is the smallest, which makes the network more robust and minimizes the impact of β. There are differences on ImageNet-100. When $\beta = 0.6$ or $\beta = 0.8$, the performance results are similar and the best. After weighing the pros and cons, we finally chose $\beta = 0.2$ and $\beta = 0.8$ as the hyperparameter settings on the two datasets.

Hash bits		32 bits	64 bits	96 bits
Cifar-100	CE	0.1384	0.1732	0.1958
	CE+CAFE	0.1741	0.2228	0.2279
	CE+CCH	0.1790	0.2275	0.2455
	ACHNet	**0.2246**	**0.2770**	**0.2957**
ImageNet-100	CE	0.6125	0.6763	0.7026
	CE+CAFE	0.7854	0.7976	0.7911
	CE+CCH	0.6245	0.6807	0.7053
	ACHNet	**0.7965**	**0.8128**	**0.8163**

Table 6: Retrieval performance to verify the effectiveness of the proposed module

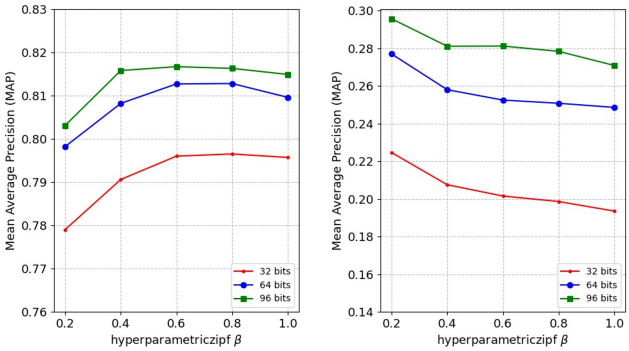

Figure 2: The retrieval results of 32bits, 64bits and 96bits under different hyperparameter settings. ImageNet-100 on the left and Cifar-100 on the right

5 Conclusion

This paper proposes an attention-guided contrastive hashing network to solve the hashing retrieval task under the long-tailed data distribution. In our method, firstly, the cross attention feature enhancement module is adopted to prevent the information loss originated by mapping high-dimensional features to hash codes. Then, the output of the hash layer is constrained by category-centered contrast learning, which solves the problem of imbalance between positive and negative sample pairs of head and tail classes. Through a large number of comparative experiments, we verify the superiority of the proposed CAFEM and CCH, while the retrieval results on six benchmarks also show that the hashing retrieval performance of ACHNet is the state-of-the-art method on long-tailed data.

Acknowledgments

Our work was supported in part by the National Natural Science Foundation of China under Grant 62132016, Grant 62171343, and Grant 62071361, in part by Key Research and Development Program of Shaanxi under Grant 2021ZDLGY01-03, and in part by the Fundamental Research Funds for the Central Universities ZDRC2102.

References

[Cai *et al.*, 2021] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 112–121, 2021.

[Cao *et al.*, 2017] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE international conference on computer vision*, pages 5608–5617, 2017.

[Cao *et al.*, 2019] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019.

[Chawla *et al.*, 2002] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[Chen *et al.*, 2021a] Yong Chen, Yuqing Hou, Shu Leng, Qing Zhang, Zhouchen Lin, and Dell Zhang. Long-tail hashing. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1328–1338, 2021.

[Chen *et al.*, 2021b] Zhi Chen, Yadan Luo, Sen Wang, Ruihong Qiu, Jingjing Li, and Zi Huang. Mitigating generation shifts for generalized zero-shot learning. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2021.

[Cui *et al.*, 2019] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[Deng *et al.*, 2018] Cheng Deng, Zhaojia Chen, Xianglong Liu, Xinbo Gao, and Dacheng Tao. Triplet-based deep hashing network for cross-modal retrieval. *IEEE Transactions on Image Processing*, 27(8):3893–3903, 2018.

[Gui *et al.*, 2017] Jie Gui, Tongliang Liu, Zhenan Sun, Dacheng Tao, and Tieniu Tan. Fast supervised discrete hashing. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):490–496, 2017.

[Kang *et al.*, 2016a] Qi Kang, XiaoShuang Chen, SiSi Li, and MengChu Zhou. A noise-filtered under-sampling scheme for imbalanced classification. *IEEE transactions on cybernetics*, 47(12):4263–4274, 2016.

[Kang *et al.*, 2016b] Wang-Cheng Kang, Wu-Jun Li, and Zhi-Hua Zhou. Column sampling based discrete supervised hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[Kang *et al.*, 2019] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2019.

[Li *et al.*, 2016] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. Feature learning based deep supervised hashing with pairwise labels. In *IJCAI*, 2016.

[Li *et al.*, 2017] Qi Li, Zhenan Sun, Ran He, and Tieniu Tan. Deep supervised discrete hashing. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2479–2488, 2017.

[Li *et al.*, 2018] Ning Li, Chao Li, Cheng Deng, Xianglong Liu, and Xinbo Gao. Deep joint semantic-embedding hashing. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2397–2403, 2018.

[Lin *et al.*, 2017] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[Liu *et al.*, 2012] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2074–2081. IEEE, 2012.

[Ren *et al.*, 2020] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *arXiv preprint arXiv:2007.10740*, 2020.

[Shen *et al.*, 2015] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 37–45, 2015.

[Wang *et al.*, 2015] Qifan Wang, Zhiwei Zhang, and Luo Si. Ranking preserving hashing for fast similarity search. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[Wang *et al.*, 2020] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2020.

[Yuan *et al.*, 2020] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3083–3092, 2020.

[Zhou *et al.*, 2020] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9719–9728, 2020.