2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

GRM: Gradient Rectification Module for Visual Place Retrieval

Boshu Lei[1], Wenjie Ding[2], Limeng Qiao[2], Xi Qiu[2]*

***Abstract*— Visual place retrieval aims to search images in the database that depict similar places as the query image. However, global descriptors encoded by the network usually fall into a low dimensional principal space, which is harmful to the retrieval performance. We first analyze the cause of this phenomenon, pointing out that it is due to degraded distribution of the gradients of descriptors. Then, we propose Gradient Rectification Module (GRM) to alleviate this issue. GRM is appended after the final pooling layer and can rectify gradients to the complementary space of the principal space. With GRM, the network is encouraged to generate descriptors more uniformly in the whole space. At last, we conduct experiments on multiple datasets and generalize our method to classification task under prototype learning framework.**

I. INTRODUCTION

Visual place retrieval (VPR) has gained much popularity in recent years. Given a query image, VPR aims to find images in the database that depict similar places. The typical pipeline of this task is to perform rough matching using global features and refine similarity using local features [1]–[3]. Most of the methods aim to obtain distinctive global and local features by improving the optimization function [4], aggregation methods [5], [6] and local branch structures [1], [2]. However, few methods consider the feature distribution, which is very important for VPR task because it directly affects the retrieval results.

Early work to address the significance of feature distribution for retrieval is [7]. The author claims that whitening can balance out the co-occurrence of features, which is beneficial to retrieval. Later, many works try to eliminate the unbalanced distribution of descriptors, like loss regularizor [8], [9] and normalization technique [10]–[12]. However, no one goes deeper into the question, what makes the model generate such unbalanced features. After thorough analysis, we attribute it to the gradients of descriptors in training process. During training, we find that both the descriptors and the corresponding gradients are trapped in the same low dimensional principal space. As updates to the descriptors, gradients trapped in principal space won't push descriptors to other dimensions, resulting in the network generating degraded features and thus harmful to the retrieval performance. We term the span of the eigenvectors with large eigenvalues from feature covariance matrix as principal space and the span of remaining vectors as complementary space.

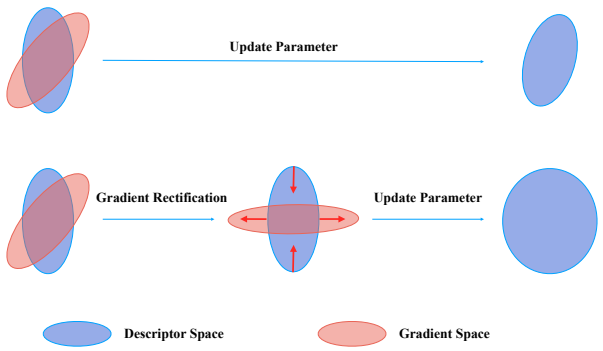

Fig. 1. **Gradient Rectification.** The circles in the figure represent the distribution of descriptors and gradients respectively. Without GRM, network generates degraded features when gradients fall into the same principal space of descriptors (Up). When GRM is applied, gradients are pulled out of this space (in orthogonal direction) and the network will generate descriptors more uniformly (Down).

We now propose to modify the distribution of descriptors through a **G**radient **R**ectification **M**odule (GRM) as illustrated in Fig. 3. First, we propose a new memory queue method to estimate the co-variance matrix of descriptors. This covariance matrix is used to generate a projection matrix to rectify gradients. Second, we propose our rectification criterion. We inhibit the component of gradients along the principal space and stimulate the component along directions orthogonal to the principal space. After doing so, gradients will be deviated from the principal space and descriptors are pulled to the complementary space of the principal space. In the forward pass, GRM stores descriptors in the memory queue for covariance estimation. In the backward pass, GRM modifies gradients based on the covariance matrix estimated from queue. It's worth noting that GRM is plug-and-play and parallel to those methods that aim to improve the loss function or local branches.

In conclusion, our contributions are threefold:

1) We point out that gradients fall into the same space of descriptors in VPR task, hindering the network from exploring other dimensions of the feature space.

2) We propose a simple yet effective Gradient Rectification Module (GRM). By modifying the gradients of descriptors, GRM pulls descriptors to the complement space of the principal space and alleviates the problem above.

3) We conduct extensive experiments in various datasets in VPR and classification tasks. Experimental results show that the proposed method significantly outperforms state of the arts.

*This work was done in MEGVII Inc.
[1]Boshu Lei is with Faculty of Automation, Xi'an Jiaotong Univerisity, Xi' an, China
[2] Wenjie Ding, Limeng Qiao and Xi Qiu are with MEGVII Inc.
* Corresponding author

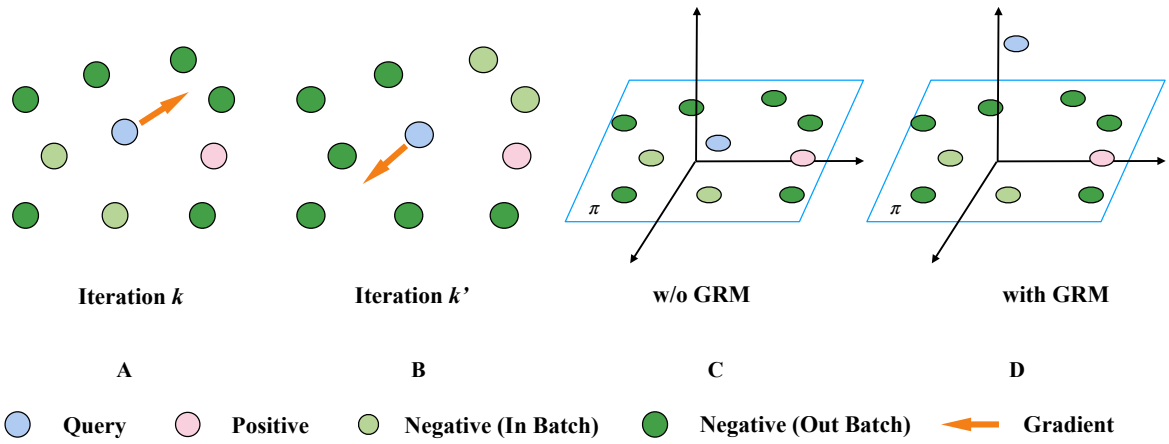

Fig. 2. **Optimization dilemma in low dimensional space.** In current step (A), the influence from the negative and positive instances will push the query instance to the direction along the arrow. In later steps (B), when negative instances in the opposite direction are sampled, they will push gradients back. We propose to pull the query out of the principal space ((C) to (D)). In that case, the query will be further away from all negative samples.

II. Related Works

A. Global Feature Descriptor

Early approaches to extract global feature descriptor are based on aggregation of hand crafted local features. Methods are like Bag of Words (BoW) [13] and Vector of Locally Aggregated Descriptors (VLAD) [14]. Sparse keypoint location or dense sampling on image are applied to aggregate local features. When deep learning technology is introduced into these frameworks, new methods like NetVLAD [5], NetBoW [15], NetFV [6] emerge. Other methods apply simple pooling layer, eg. max pooling or GeM pooling [16], on top of the CNN architecture [4], [12]. These methods are trained with Triplet Loss [5], or ranking-based loss [17], [18]. Most recently, [4] proposes to use a continuous measure to calculate the similarity between images. Instead of depending on the binary relation, they use the calculated similarity score to train the neural network and achieve huge success in MSLS challenge [19].

B. Local Feature Refinement

Early methods for refinement are separated from global feature matching scheme. Local features like SIFT [20], SURF [21] are extracted and matched based on nearest neighbor criterion. Then model-based methods, such as RANSAC [22] and PROSAC [23] are used to generate transformation hypotheses based on correspondences. The final retrieval result is re-ranked by the number of inliers under the hypothesis. DELF [3] proposes to use the feature map before pooling for later refinement process. PatchNetVLAD [1] proposes to extract NetVLAD [5] descriptors on image patches for later refinement procedure. [2] is the first to jointly learn local and global features for retrieval task.

C. Whitening and Spreading Methods

According to [24], when the features become white, the Fisher information matrix reduces to an identity matrix. In that case, the simple stochastic gradient descent optimization coincides with the natural gradients descent optimization which is more suitable for difficult optimzation tasks. Whitening methods can be classified into two categories. The first category is to add extra penalty terms in loss function. Zhang et al. [8] proposed to add a regularizor to spread vectors uniformly in the feature space. In [9], the author provides a thorough analysis of the uniformity and alignment metric of the feature distribution space. Additional loss penalties are added to adjust the uniformity and alignment of feature space. Another method is to process the features directly, either in the forward pass or the backward pass. Batch Normalization (BN) [10] is the most widely used method in the forward pass of deep learning model. Later, [11] constructs the whitening matrix based on zero-phase component analysis to mitigate the stability issue in BN and PCA [25]. [12] appends a PCA projection layer behind the network to spread descriptors. [26] finds that the whitening matrix exerted on features can be instead applied on model weights. They prove that their methods are mathematically equivalent to ZCA [11] but the computation complexity is largely reduced. WADAM [27] extends the idea in [26] with momentum, adaptive dampening and proposes new optimizers for network training.

In VPR, [7] points out that whitening can balance out the co-occurrence of features, which is generally beneficial to retrieval tasks. Whitened features can boost the performance of the network on the current datasets and improve the generalization ability of the network across datasets [4]. However, Whitening adds a strong prior on the distribution of the descriptors and is not involved in the training process.

Our work focuses on the global descriptor extraction procedure, since the quality of the global descriptor is the bottleneck of later local descriptor matching process. Unlike previous works in VPR [4], [5], [17], [28], we do not propose new aggregation methods or new loss functions, but instead focus on the feature distribution that lacks attention. We propose a new method to optimize the feature distribu-

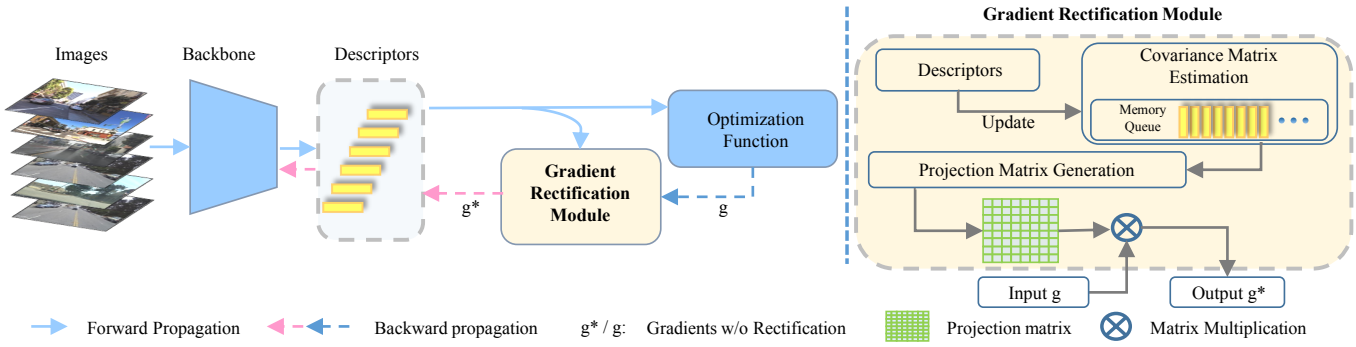

Fig. 3. **The learning paradigm of the proposed method.** Forward descriptors are accumulated in the memory queue and later used to generate the projection matrix. Backward gradients are multiplied by this matrix when flowing through the GRM.

tion, which is parallel to most VPR methods before-head.

III. Method

A. Gradient Distribution Problem

We claim that pair-wise loss function in VPR is one of the reasons for the degraded distribution of gradients. A general pair-wise loss function can be written as

$$L = \sum_{i} f(s_{i,1}, ..., s_{i,N}) \tag{1}$$

where $s_{i,j}$ is the similarity between query i and sample j. It is usually defined as cosine similarity or L2 distance. These two are equivalent when the descriptors are normalized. For simplicity, we will use the L2 distance.

$$s_{i,j} = ||\vec{p_i} - \vec{p_j}||^2, \quad \vec{p_i} \in R^C \tag{2}$$

$\vec{p_i}$ and $\vec{p_j}$ are the descriptors encoded by the network. The gradient of the descriptor is

$$\frac{\partial L}{\partial \vec{p_i}} = \sum_{j=1}^{N} \frac{\partial f}{\partial s_{i,j}} 2(\vec{p_i} - \vec{p_j}) \tag{3}$$

Since $2\frac{\partial f}{\partial s_{i,j}}$ is a scalar, Eq. 3 can be written as

$$\frac{\partial L}{\partial \vec{p_i}} = \sum_{j=1}^{N} \alpha_j (\vec{p_i} - \vec{p_j}) \tag{4}$$

Here, we use contrastive loss as an example. The contrastive loss (CL) is shown in Eq. 5.

$$L_{CL}(\vec{p_i}) = \sum_{j=1}^{N} \phi_{i,j} s_{i,j} + (1 - \phi_{i,j}) max(\tau - s_{i,j}, 0) \tag{5}$$

$\phi_{i,j} = 1$ if sample i and j are positive pair and 0 otherwise. τ is the threshold. The derivative is

$$\frac{\partial L}{\partial \vec{p_i}} = \sum_{j=1}^{N} [\phi_{i,j} - (1 - \phi_{i,j}) sign(\tau - s_{i,j})](\vec{p_i} - \vec{p_j}) \tag{6}$$

In Eq. 6, since $\phi_{i,j} - (1-\phi_{i,j}) sign(\tau - s_{i,j})$ is a scalar, the gradient is a linear combination of the query and database descriptors. Since the projection of descriptors in the principal space is much larger than the projection in the complement space, a linear combination of these principal components is generally larger than other components. Therefore, gradients will also have large components in the principal space.

B. Gradient Rectification Module

The whole process can be summarized in Fig. 3. During training, we feed images into the network to get descriptors. These descriptors are used to update the estimated covariance matrix P. Loss is computed according to the learning task, for instance, GCL [4] or Triplet Loss. Then we backward the loss and capture gradients at the descriptor level. The projection matrix P^* is applied here and the modified gradients flow into the network.

1) Covariance Matrix Estimation: A brutal way to estimate the covariance matrix is to feed all training examples into the network and calculate the covariance matrix with them after each epoch. This can be applied when the size of the dataset is relatively small. However, when the size of the dataset is large, this operation is time consuming. What is worse, since the descriptors are usually designed to be high dimensional, the number of examples inside one batch is inadequate to provide an accurate estimation of the covariance matrix, if using the regular formula below:

$$P = \frac{1}{B-1} \sum_{i=1}^{B} (\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^T. \tag{7}$$

Here, P is the estimated covariance matrix. B is the batch size. $\boldsymbol{x}$ is the descriptor and $\bar{\boldsymbol{x}}$ is the mean descriptor within the current batch.

Therefore, non-singularity and positive-definite property of the covariance matrix may not be preserved. To this end, we borrow the idea, memory queue, from unsupervised learning domain [29]. We manage a queue of size K. The current batch is enqueued to the memory queue and the oldest batch in the queue is removed. The queue represents a sampled subset of all the instances. Covariance matrix are

estimated using the standard form from instances in memory queue. In order to prevent numerical instability, we also add a small number (10^{-3}) to each of the diagonal elements of P.

2) Projection Matrix Generation: We apply eigenvalue decomposition on P:

$$P = U diag(\lambda_1, ..., \lambda_n) U^T \tag{8}$$

$\lambda_1, ..., \lambda_n$ are eigenvalues in descending order. Then we form the projection matrix P^* to modify gradients.

$$P^* = U\, diag(\frac{\bar{\lambda}}{\lambda_1}, ..., \frac{\bar{\lambda}}{\lambda_n})^s U^T \tag{9}$$

$\bar{\lambda}$ is the mean of the eigenvalues. To eliminate the scale problem, we put $\bar{\lambda}$ instead of a constant on the numerator. Since at the initial stage of the training process, the elements in the descriptors are all near zero, causing all the eigenvalues to be very small. A constant in the numerator may cause a large scaling value along this dimension, leading to unstable training at the beginning. s is a hyper-parameter controlling the rectification rate. Large $s(> 1)$ causes big rectifying value and leads to unstable training while small $s(< 0.5)$ is unable to push the network to generate descriptors uniformly. We manually set this parameter to 1.

3) Gradient Rectification: We rectify gradients using projection matrix P^* as follows

$$g^* = P^* g \tag{10}$$

Modified gradients g^* flows into the backbone network. Next, optimizers like SGD or Adam [30] are adopted to update network's parameters. In the ideal case, when the covariance matrix of gradients is only different from the covariance matrix of the descriptors by a constant multiplier, it is trivial to show that the variance of gradients is identical along each dimension.

The space complexity of GRM is $(K + C)C$, where C is the dimension of descriptors. KC is the size of the memory queue and C^2 is the size of the covariance matrix. The time complexity for covariance matrix estimation is $O(KC^2)$ and for eigenvalue decomposition is $O(C^3)$. GRM is not involved in the inference stage, it doesn't impact inference speed of the model.

C. Datasets

1) MSLS: MSLS is a large scale place recognition dataset that contains images taken in 30 different cities [19]. The training set contains over 500k query images and 900k map images. The validation set consists of 19k map images and 11k query images and the test set has 39k map images and 27k query images. Two images are considered as similar if they are taken by cameras located within 25m of distance and with less than 40° of viewpoint variation.

2) Pittsburg: This dataset contains images of urban environments gathered from google street view in the city of Pittsburgh, Pennsylvania, USA [5]. We use the test set of Pitts30K to test our model. The test set of Pitts30K contains 6k query images and 10k map images.

3) Tokyo 24/7: It consists of images taken in Tokyo, Japan, with large variations of illumination, as they are taken during day and night [31]. The query set consists of 315 images and the map contains 76k photos.

4) Nordland: This dataset is recorded on a train at four different seasons [32]. Each image is taken at the same place as corresponding ones in other seasons. We use the test subset, which consists of 3.5k images in each season, to test our model. The tolerance for positive match is set to 10 frames and we use images in summer as queries and images in winter as databases, as PatchNetVLAD [1].

D. Implementation Details

We use GCL Loss [4] to train our model. We select ResNet50 [33] and ResNext101 [34] as backbone to extract features. GeM pooling is applied to get the global descriptor. The memory queue size (K) is 10,240. We use Adam optimizer [30] and set the learning rate to 1e-4. We train our model for 200 epochs (10k pairs in MSLS per epoch). Following GCL [4], we use the weights pretrained on ImageNet and only train the last two blocks of the backbone. For GRM, the rectification rate is set to 1. We train our network on MSLS training set and test them on other datasets. Top-N recall (R@N) is adopted as evaluation metric.

E. Comparison with State-of-the-art Methods.

For a fair comparison, we only compare the first stage of all methods without the re-ranking process. We compare the performance of GRM with state of the arts in Table I. On MSLS dataset, even though the baseline methods, GCL-ResNet50 and GCL-ResNext101, are superior than all the other methods, our methods can still boost the performance. ResNext101 trained by our method can achieve the best R@5 in the official MSLS challenge, at 77.2%. For ResNet50, the R@5 increases nearly 10%, and is comparable to the larger ResNext101 model. Our methods can also bring huge improvement on Nordland dataset. This is due to the fact that season variation consists a large portion of the MSLS dataset. Therefore, our methods trained on MSLS generalize well on Nordland dataset, increasing over 5% on ResNext101 and over 30% on ResNet50. In urban environment, Pittsburg 30K and Tokyo24/7, our method can bring moderate improvement on the baseline method, between 1˜2%, compared to the baseline method. Our best R@5 on Pittsburg30K (91.9%) and Tokyo24/7 (75.2%) is lower than PatchNetVLAD (94.5% on Pittsburg30K and 88.6% on Tokyo24/7). This is because Patch-NetVLAD is fine-tuned on Pittsburg30K training dataset, with images including drastic variations in illumination. However, in order to keep aligned with other methods, we only train our method on MSLS dataset.

F. Comparison with Other Whitening Methods

We compare the performance of GRM and other feature whitening methods. We select GCL [4] as our baseline and use multiple strategies to expand the feature space. For

TABLE I

COMPARISON WITH STATE-OF-THE-ART METHODS.

Recall (%)	MSLS (Test. set)			MSLS (Val. set)			Pittsburgh 30k			Nordland			Tokyo247		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
NetVLAD 64 [5]	45.1	58.8	63.7	70.1	80.8	84.9	68.6	84.7	88.9	-	-	-	34.0	47.6	57.1
NetVLAD 16 [5]	39.4	53.0	57.5	70.5	81.1	84.3	70.3	84.1	89.1	-	-	-	37.8	53.3	61.0
Patch-NetVLAD [1]	48.1	57.6	60.5	79.5	86.2	87.7	**87.7**	**94.5**	**95.9**	71.3	80.9	82.2	**86.0**	**88.6**	**90.5**
GCL (ResNet50) [4]	52.9	65.7	71.9	74.6	84.7	88.1	79.9	90.0	92.8	44.7	58.8	65.3	58.7	71.7	76.8
GCL (ResNext101) [4]	62.3	76.2	81.1	80.9	90.7	92.6	79.2	90.4	93.2	69.9	85.6	90.1	58.1	74.3	78.1
TransVPR w/o. rerank [28]	48.0	67.1	73.6	70.8	85.1	89.6	73.8	88.1	91.9	-	-	-	-	-	-
Ours (ResNet50)	62.3	75.5	78.7	82.4	90.2	91.7	82.8	91.9	93.9	66.2	89.9	93.0	55.9	70.8	76.2
Ours (ResNext101)	**64.9**	**77.2**	**81.1**	**82.6**	**90.7**	**92.6**	81.3	91.4	93.5	**82.3**	**92.4**	**95.2**	62.5	75.2	82.5

Spread Loss [8]. We set the weight for the regularization term to 1. For AP Loss [17]. We use the GCL Label in [4] to sample 2 positive images and 7 negative images. Negative images are half hard negative (similarity score greater than 0 but less than 0.5) and half negative (similarity score equals 0). We compare our method to [27]. Standard Adam [30] optimizer is replaced by WADAM [27] in the GCL [4] framework.

The training procedure follows the previous section and we show the R@N index on MSLS val and test sets in Table II. GRM exceeds all methods in Table II. The performance drops when adding the extra regularization in [8]. We conjecture that the inclusion of regularization might break the original distribution and leads to a sub-optimal distribution. As we can see, AP Loss [17], as the representation of multi-pair form, increase R@5 by 4% on MSLS test set but is still lower than our method. Meanwhile, AP Loss suffers from large memory consumption and long training time. As for the WADAM [27], it can bring slight improvement on MSLS val., but when evaluated on test set, it under-performs the baseline method.

The result corroborates that it is neither the weak repulsive force nor the limited batch size that causes the unbalanced distribution of descriptors but the phenomenon that gradients and descriptors fall into the same low dimensional principal space. Since gradients won't push the descriptors out of this principal space, descriptors will dwell in this principal space and never be uniformly distributed.

TABLE II

COMPARISON WITH WHITENING METHOD.

Recall (%)	MSLS (Test. set)			MSLS (Val. set)		
	R@1	R@5	R@10	R@1	R@5	R@10
ResNet50 [4]	52.9	65.7	71.9	74.6	84.7	88.1
Spread Loss [8]	30.8	47.2	53.6	49.1	63.4	68.6
AP Loss [17]	53.4	69.7	75.0	72.3	82.4	84.9
WADAM [27]	48.4	62.6	68.7	72.9	85.1	87.9
ResNet50 + GRM	**62.3**	**75.5**	**78.7**	**82.4**	**90.2**	**91.7**

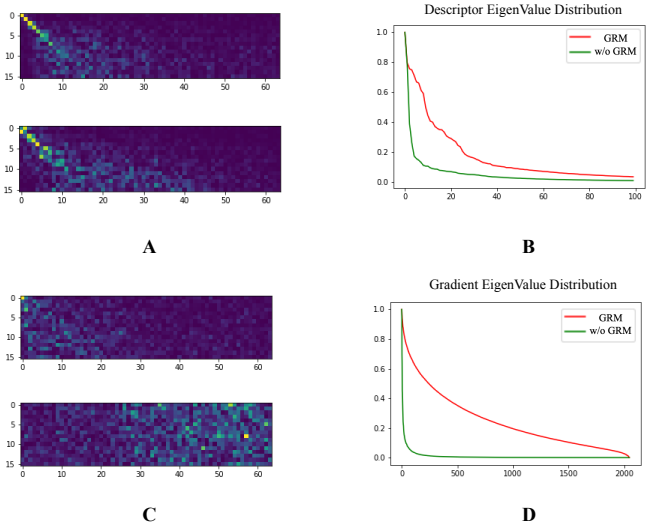

Fig. 4. **Gradient and descriptor distribution.** (A) and (C) are alignment of eigenvectors between different training stage w/o. GRM and with GRM. (B) and (D) are the eigenvalue distribution of descriptors and gradients with and w/o. GRM.

G. Ablation Study

1) Gradient distributions analysis: In order to illustrate the problem that the descriptors encoded by the network fall into a low dimensional space and do not change during the training process, we plot the correlation between the principal space of descriptors in different stages during training process in Fig. 4. In the upper figure of (A), pixel value at (i, j) is the dot product between the i-th eigenvetor of the feature space encoded by model after training 30 epochs and the j-th eigenvector of the feature space encoded by model after training 90 epochs, **without GRM**. Here, all the eigenvectors are sorted by their corresponding eigenvalue. The i-th eigenvector is the eigenvector corresponds to the i-th largest eigenvalue. Therefore, the brighter the pixel is, the more aligned two eigenvectors are. In the lower figure of (A), pixel value at (i, j) is the dot product between the i-th eigenvetor of the feature space and the j-th eigenvector of the gradient space. Fig. 4 (C) is the same as (A) but the model is trained **with GRM**.

In Fig. 4 (A), We notice that brightest pixels are on the

diagonal line of the image, indicating that eigenvectors are aligned respectively during training process and the principle space does not change. The same situation remains between the principle space of descriptors and gradients. Given that descriptors and gradients all concentrate in a low principal space (the green curve in Fig. 4 B and D), we claim that descriptors trapped in a low dimensional space is the result of the gradients which fall into the same dimensional space.

When GRM is applied, bright pixels disperse in Fig. 4 (C). Hence the principal space between descriptors and gradients are not aligned. Gradients are pulled out of the principal space of descriptors and influence the distribution of descriptors in later steps. We also plot the eigenvalue of the descriptor space and the gradient space in Fig. 4 (B) and (D). We can find that the descriptors and gradients distribute more uniformly when GRM (red curve) is applied.

2) Experiments in Classification Tasks.: To evaluate the performance of the proposed method on classification tasks, we test our method on CIFAR-10, CIFAR-100, Caltech 101 and Caltech 256 datasets [35]. We adopt the prototype learning method in [36]. We use ResNet18 as backbone to extract features and use average pooling to get the descriptor. The feature dimension is set to 512. We use the GCPL loss in [36] to train our network. The gradient rectification module is applied to descriptors and prototypes simultaneously. The model is optimized using SGD with momentum of 0.9. The initial learning rate is set to 0.05 and is multiplied by 0.7 every 20 epochs. We initialize all prototypes as zero [36] to avoid noise. The whole training process lasts for 200 epochs. We use the top-1 accuracy to evaluate all the models. Results are summarized in Table III.

GRM can bring improvement in prototype learning framework. It brings around 0.5% improvement on CIFAR-10 and CIFAR-100 using the GCPL framework. On Caltech dataset, even though the GCPL framework [36] underperforms the baseline method, our method can still achieve better results, about 1˜2% than the baseline method.

TABLE III

PERFORMANCE ON CLASSIFICATION TASK.

Accuracy (%)	CIFAR-10	CIFAR-100	Caltech101	Caltech256
ResNet18 (Baseline)	93.2	73.3	74.3	73.5
ResNet18 + GCPL	94.4	75.5	71.5	71.4
ResNet18 + GCPL + **GRM**	**94.9**	**75.8**	**75.2**	**75.3**

3) Comparison with Other Estimation Method: Another way to estimate the covariance matrix is running average. The procedure is given below:

$$
\begin{aligned}
N_{k+1} &= N_k + b \\
\bar{x}_{k+1} &= \frac{N_{k+1}-b}{N_{k+1}}\bar{x}_k + \frac{b}{N_{k+1}}\sum_{i=1}^{b} x_i \\
P_{k+1} &= \frac{N_{k+1}-b}{N_{k+1}}P_k + \frac{b}{N_{k+1}}\sum_{i}^{b}(x_i-\bar{x}_{k+1})(x_i-\bar{x}_{k+1})^T
\end{aligned}
\tag{11}
$$

b is the batch size. k is the number of step. x is the descriptor. P is the co-variance matrix. $\bar{x}$ is the running mean. P is initialized as identity and $\bar{x}$ is initialized as zero. In each step, the projection matrix is calculated from P_k and multiplies the backward gradients. We use the same setting to test the running average method and compare it with baseline and memory queue method.

The running average method can still outperform the baseline method on MSLS. It gets a top-5 recall rate of 73.2% on MSLS test set, higher than the baseline (65.5%), which is still significant. On MSLS validation set, it gets top-5 recall at 88.4%, 4% higher than the baseline method. It is even slightly better on Pittsburg30K than the memory queue method. Moreover, running average method doesn't need to maintain a large memory queue to store descriptors, cutting off the memory consumption.

TABLE IV

COMPARISON WITH RUNNING AVERAGE METHOD.

method	MSLS (Test. set)			MSLS (Val. set)		
	R@1	R@5	R@10	R@1	R@5	R@10
GCL (ResNet50) [4]	52.9	65.7	71.9	74.6	84.7	88.1
ResNet50 + Average (Sqrt)	61.5	73.2	77.2	**84.0**	88.7	90.7
ResNet50 + Bank (Linear)	**62.3**	**75.5**	**78.7**	82.4	**90.2**	**91.7**

4) Memory Queue Size: Here we conduct experiments to analyze the effect of memory queue size K. The result is summarized in Table V. We find that small queue will harm the performance of the model. This is because the estimated covariance matrix is not accurate from few samples. Queue size larger than 10,000 suffices an accurate estimation.

TABLE V

MEMORY QUEUE SIZE COMPARISON

Memory Queue Size	MSLS (Test. set)			MSLS (Val. set)		
	R@1	R@5	R@10	R@1	R@5	R@10
GCL (ResNext101) [4]	62.3	76.2	81.1	80.9	90.7	92.6
K=5120	59.1	71.4	75.7	80.4	87.6	89.9
K=7680	62.4	73.8	76.8	81.9	89.1	90.6
K=10240	**64.9**	**77.2**	**81.1**	**82.6**	**90.7**	**92.6**

IV. CONCLUSIONS

In this article, we point out the gradient problem in training deep neural network for global feature extraction in VPR. The degraded distribution of gradients leads to the unbalanced distribution of descriptors. We propose our new GRM and memory queue method to rectify gradients. This method can improve the existing state of the art methods on global descriptor extraction. We also demonstrate that our method could be generalized to classification task. For future work, the GRM Module can be applied to other metric learning tasks like unsupervised learning or re-ID in tracking task.

REFERENCES

[1] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 14 141–14 152. I, II-B, III-C.4, I

[2] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12365. Springer, 2020, pp. 726–743. [Online]. Available: https://doi.org/10.1007/978-3-030-58565-5_43 I, II-B

[3] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 3476–3485. [Online]. Available: https://doi.org/10.1109/ICCV.2017.374 I, II-B

[4] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, "Generalized contrastive optimization of siamese networks for place recognition," *CoRR*, vol. abs/2103.06638, 2021. [Online]. Available: https://arxiv.org/abs/2103.06638 I, II-A, II-C, III-B, III-D, III-F, I, II, IV, V

[5] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: CNN architecture for weakly supervised place recognition," *CoRR*, vol. abs/1511.07247, 2015. [Online]. Available: http://arxiv.org/abs/1511.07247 I, II-A, II-B, II-C, III-C.2, I

[6] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," *CoRR*, vol. abs/1706.06905, 2017. [Online]. Available: http://arxiv.org/abs/1706.06905 I, II-A

[7] H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening," in *ECCV*, 2012. I, II-C

[8] X. Zhang, F. X. Yu, S. Kumar, and S. Chang, "Learning spread-out local feature descriptors," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 4605–4613. [Online]. Available: https://doi.org/10.1109/ICCV.2017.492 I, II-C, III-F, II

[9] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2495–2504. I, II-C

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, ser. JMLR Workshop and Conference Proceedings, F. R. Bach and D. M. Blei, Eds., vol. 37. JMLR.org, 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html I, II-C

[11] A. Kessy, A. Lewin, and K. Strimmer, "Optimal whitening and decorrelation," *The American Statistician*, vol. 72, no. 4, pp. 309–314, 2018. I, II-C

[12] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9910. Springer, 2016, pp. 241–257. [Online]. Available: https://doi.org/10.1007/978-3-319-46466-4_15 I, II-A, II-C

[13] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2. II-A

[14] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. IEEE Computer Society, 2010, pp. 3304–3311. [Online]. Available: https://doi.org/10.1109/CVPR.2010.5540039 II-A

[15] E. Ong, S. Husain, M. Bober-Irizar, and M. Bober, "Deep architectures and ensembles for semantic video classification," *CoRR*, vol. abs/1807.01026, 2018. [Online]. Available: http://arxiv.org/abs/1807.01026 II-A

[16] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018. II-A

[17] J. Revaud, J. Almazán, R. S. Rezende, and C. R. de Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 5106–5115. [Online]. Available: https://doi.org/10.1109/ICCV.2019.00521 II-A, II-C, III-F, II

[18] F. Çakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 1861–1870. II-A

[19] F. Warburg, S. Hauberg, M. López-Antequera, P. Gargallo, Y. Kuang, and J. Civera, "Mapillary street-level sequences: A dataset for lifelong place recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 2623–2632. II-A, III-C.1

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94 II-B

[21] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. II-B

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8. II-B

[23] O. Chum and J. Matas, "Matching with prosac - progressive sample consensus," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 220–226 vol. 1. II-B

[24] G. Desjardins, K. Simonyan, R. Pascanu, *et al.*, "Natural neural networks," *Advances in neural information processing systems*, vol. 28, 2015. II-C

[25] L. Huang, D. Yang, B. Lang, and J. Deng, "Decorrelated batch normalization," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 791–800. II-C

[26] S. Markovich-Golan, B. Battach, and A. Bleiweiss, "Feature whitening via gradient transformation for improved convergence," *CoRR*, vol. abs/2010.01546, 2020. [Online]. Available: https://arxiv.org/abs/2010.01546 II-C

[27] H. Yong and L. Zhang, "An embedded feature whitening approach to deep neural network optimization," in *the European Conference on Conputer Vision*, 2022. II-C, III-F, II

[28] R. Wang, Y. Shen, W. Zuo, S. Zhou, and N. Zheng, "Transvpr: Transformer-based place recognition with multi-level attention aggregation," *CoRR*, vol. abs/2201.02001, 2022. [Online]. Available: https://arxiv.org/abs/2201.02001 II-C, I

[29] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 9726–9735. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.00975 III-B.1

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980 III-B.3, III-D, III-F

[31] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 257–271, 2018. [Online]. Available: https://doi.org/10.1109/TPAMI.2017.2667665 III-C.3

[32] D. Olid, J. M. Fácil, and J. Civera, "Single-view place recognition under seasonal changes," *CoRR*, vol. abs/1808.06516, 2018. [Online]. Available: http://arxiv.org/abs/1808.06516 III-C.4

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90 III-D

[34] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 5987–5995. [Online]. Available: https://doi.org/10.1109/CVPR.2017.634 III-D

[35] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, 2006. [Online]. Available: https://doi.org/10.1109/TPAMI.2006.79 III-G.2

[36] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Robust classification with convolutional prototype learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3474–3482. III-G.2