Posterior Refinement on Metric Matrix Improves Generalization Bound in Metric Learning

Mingda Wang[1], Canqian Yang[1], and Yi Xu[1] *

MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
{597924594,charles.young,xuyi}@sjtu.edu.cn

Abstract. Deep metric learning (DML) attempts to learn a representation model as well as a metric function with a limited generalization gap, so that the model trained on finite known data can achieve similitude performance on infinite unseen data. While considerable efforts have been made to bound the generalization gap by enhancing the model architecture and training protocol a priori in the training phase, none of them notice that a lightweight posterior refinement operation on the trained metric matrix can significantly improve the generalization ability. In this paper, we attempt to fill up this research gap and theoretically analyze the impact of the refined metric matrix property on the generalization gap. Based on our theory, two principles, which suggest a smaller trace or a smaller Frobenius norm of the refined metric matrix, are proposed as guidance for the posterior refinement operation. Experiments on three benchmark datasets verify the correctness of our principles and demonstrate that a pluggable posterior refinement operation is potential to significantly improve the performance of existing models with negligible extra computation burden.

Keywords: deep metric learning, generalization, metric matrix, posterior refinement

1 Introduction

Deep metric learning (DML) attempts to map instances onto an embedding space, in which similar instances are closer to each other by means of a predefined distance metric function. The most studied metric is Mahalanobis distance, which is parameterized by a metric matrix learned automatically from the data. To enhance the discriminability of the learned metric, recent works focus on the constraints on the embedding space, such as the loss functions [6, 27, 26, 14, 21, 11, 20] which provide direct criterion to learn powerful embedding space, the mining strategies [29, 7, 9, 25] which select training samples contributing significantly to the training procedure, and topology-based methods [30, 32] which considers prior knowledge about the data manifold.

* Corresponding Author

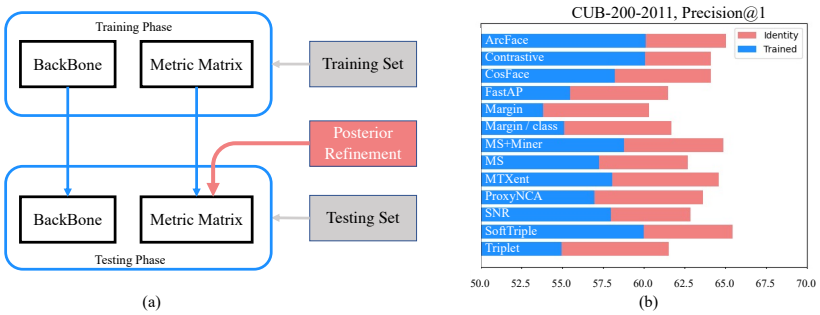

Fig. 1: (a) Demonstration about the standard pipeline of training and testing phase for DML models (in blue boxes and arrows). Posterior refinement (red box) can serve as a data-free pluggable operation on the trained metric matrix before the testing phase. (b) Comparison between the standard pipeline (evaluated by the *trained* metric directly) and refining the metric matrix to one *identity* metric matrix.

In the common scenarios of DML, there is no overlap in the data category between the training set and the testing set, which implies extreme data distribution shift. Therefore, achieving good generalization ability of the learned metric is one crucial problem for DML. The generalization ability can be quantified by the generalization gap, which is the difference between the evaluation error on the training set (called *empirical risk error*) and the whole space of possible data (called *expected risk error*), measures the capacity that the metric learned on the finite training set can yield approximate performance on infinite unseen data. The generalization gap is proven to be influenced by the training hypothesis [4] and model magnitude [8]. Accordingly, several methods with theoretical guarantee and practical achievement, including more effective training strategies [4, 13] and model regularizer [2, 18], are proposed to provide prior guidance **during the training phase** to improve the generalization ability of the trained model.

Despite the efforts mentioned above to learn a more discriminative and generative metric, we uncover an important fact that, even though trained by state-of-the-art DML methods, the trained metric matrix still generalizes no better than an identity matrix as shown in Fig. 1(b). Compared to the standard evaluation pipeline of DML, evaluating with identity metric matrix can be viewed as an additional operation that refines the learned metric matrix before the testing phase as shown in Fig. 1(a) (in this case, the learned metric matrix is refined to be an identity matrix). Such discovery implies that there exists a certain refinement operation on the learned metric matrix which can reduce the generalization gap efficiently. We call these operations as *posterior refinement*, since they are data-free methods that enhance the property of the metric matrix **after the training phase**. As far as we know, there is no existing work investigating in this subject. Therefore, in this paper, we try to fill up this research gap and provide a theoretical explanation of our discovery. To this end, we establish an

upper bound of the generalization gap, which suggests that a smaller trace or smaller Frobenius norm of the refined metric matrix facilitates the generalization ability and vice versa. Based on our formula of the generalization gap, two principles are proposed to serve as the foundation of future posterior refinement operations. The contributions of this paper are three-fold. (1) We indicate the fact that posterior refinement on the metric matrix can improve generalization ability in DML. (2) We provide two principles, which suggest a smaller trace or smaller Frobenius norm of the refined metric matrix, to guide the conduction of the posterior refinement operation and corresponding theoretical analysis to support these principles. (3) We conduct comprehensive experiments to verify the correctness of the proposed principles and demonstrate the effectiveness of the posterior refinement operations.

2 Preliminaries

2.1 Notation

Let $D_n = \{(I_1, y_1), (I_2, y_2), \ldots, (I_n, y_n)\}$ denote a dataset with n instances, where $I_i \in \mathcal{I}$ is the input image sampled from an unknown distribution space $\mathcal{I}$ and y_i is the category label of I_i. For simplicity, we assume that each category contains m instances and there are totally n/m different categories in D_n. Let f denote a feature extractor model parameterized by θ, which maps an image to a d dimensional embedding vector x_i, denoted as $x_i := f(I_i, \theta)$. We assume that x_i is independently and identically distributed (i.i.d.) sampled from an unknown distribution space, $x_i \in \mathcal{X} \subseteq \mathbb{R}^d$. Let $X_n = [x_1, x_2, \ldots, x_n]$ denote the matrix of embedding vectors extracted from the images in D_n, then $X_n \in \mathcal{X}^n \subseteq \mathbb{R}^{d \times n}$. Let $M \in \mathcal{M} \subset \mathbb{R}^{d \times d}$ denote a trainable metric matrix, where $\mathcal{M}$ denotes the space of symmetric positive semi-definite matrices.

Given a dataset D_n, a feature extractor model f and a metric matrix M, the evaluation error of DML model, denoted as $L(X_n, M)$, can be derived as follows. Let $sim(x_i, x_j, M)$ denote the similarity between the embedding vectors of two instances under the metric of M. Let $\mathcal{T}$ denote the set of all possible data triplets collected from D_n, where each data triplet $t = \{i, j, k\} \in \mathcal{T}$ is composed of the instance index of an anchor instance i, a positive instance j and a negative instance k. $sim(x_i, x_j, M)$ is supposed to overpass $sim(x_i, x_k, M)$. In practice, the positive instance is generally sampled from the instance set sharing the same category with the anchor, and the negative instance belongs to another different category. Let $l(t, M) = max\{sim(x_i, x_k, M) - sim(x_i, x_j, M), 0\}$ denote the error caused by one data triplet. When $l(t, M) = 0$, the similarity between positive data pairs has surpassed that of negative pairs so there is no error. Otherwise, a bigger $l(t, M)$ implies a relatively higher degree that the model deviates from the successful discrimination of data triplet t. Then, the evaluation error $L(X_n, M)$ can be expressed as the expected error for all data triplets

$$L(X_n, M) = \mathop{\mathbb{E}}_{t \in \mathcal{T}}[l(t, M)] = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} l(t, M) \tag{1}$$

where $|\mathcal{T}|$ denotes the number of data triplets in $\mathcal{T}$.

Now we rewrite Eq.1 in the form of matrices. Let $S = X_n^T M X_n$ denote the pairwise similarity matrix of X_n, where each element of S is equal to the similarity of two instances, $s_{ij} = x_i^T M x_j = sim(x_i, x_j, M)$. For each data triplet $t = \{i, j, k\}$, a $n \times n$ sparse sampling matrix C^t is generated, where $c_{ji}^t = -1$, $c_{ki}^t = 1$ and all other elements are zero. Let $A := SC^t$, so the diagonal elements of matrix A are all zero except $a_{ii} = sim(x_i, x_k, M) - sim(x_i, x_j, M)$. Let Λ^t denote a sparse matrix where $\lambda_{ii} = 1$ if $a_{ii} > 0$ and $\lambda_{ii} = 0$ otherwise. Therefore, $l(t, M)$ can be rewritten in the form of matrices as $l(t, M) = tr(A\Lambda^t) = tr(X_n^T M X_n C^t \Lambda^t)$.

However, consider one case that $M_1 = 2M_2$, then $l(t, M_1) = 2l(t, M_2)$, but the performance of M_1 and M_2 will be exactly the same because the similarity ranking of data pairs are not changed. To eliminate the influence of the scalar on the metric matrix, we introduce $P(M)$ to $l(t, M)$, defining the error as $l(t, M) := tr(X_n^T M X_n C^t \Lambda^t)/P(M)$, where $P(M)$ represents a certain property of the metric matrix. It is easy to verify that in the case of $M_1 = 2M_2$, if we define the property function as the Frobenius norm $P(M) := \|M\|_F$, then $l(t, M_1) = l(t, M_2)$. Thus, in this paper we will only discuss the reduction of generalization gap **on condition that** $P(M)$ **is fixed**.

Then, $L(X_n, M)$ can be rewritten as

$$\begin{aligned} L(X_n, M) &= \frac{1}{P(M)|\mathcal{T}|} \sum_{t \in \mathcal{T}} tr(X_n^T M X_n C^t \Lambda^t) \\ &= \frac{1}{P(M)|\mathcal{T}|} tr(X_n^T M X_n (\sum_{t \in \mathcal{T}} C^t \Lambda^t)) \\ &= \frac{1}{P(M)|\mathcal{T}|} tr(X_n^T M X_n C_n) \end{aligned} \tag{2}$$

where $C_n := \sum_{t \in \mathcal{T}} C^t \Lambda^t$ is named as the *similarity sampling matrix* for D_n.

In the general protocol of DML, two datasets sharing non-overlap in class category are collected, namely the training set D_{tr} and the testing set D_{te}. For simplicity, in this paper the instance numbers of D_{tr} and D_{te} are both assumed to be n and each category is assumed to contain m instances, which is generally the case of DML benchmark datasets as described in Section 4.1. This assumption means that $\mathcal{T}$ and $|\mathcal{T}|$ are exactly the same for D_{tr} and D_{te}. Therefore, in the rest of this paper we will not distinguish these symbols separately for the training set and testing set.

2.2 Generalization gap of DML

Given a training set D_{tr} and a metric matrix M, the empirical risk error is defined as the evaluation error of the current model on D_{tr}, which can be represented as $empirical \quad risk \quad error := L(X_{tr}, M)$. According to the empirical risk minimization (ERM) optimization principle, the task of the DML training

procedure is to obtain optimal parameters for feature extractor function θ^* and an optimal metric matrix M^* such that

$$\theta^*, M^* = \arg\min_{\theta, M} L(X_{tr}, M)$$

The training procedure of the DML model attempts to minimize the empirical risk error via a certain optimization algorithm, such as gradient descent (GD).

Since we concern more about the performance of the trained model on unseen data, the expected risk error, which refers to the evaluation error on the whole space of all possible data, is wished to approximate the empirical error which has been reduced during the training procedure. Therefore, a good DML model should have a small generalization gap, which is the difference between the empirical risk error and expected risk error. Let $L(\mathcal{X}, M^*)$ denote the expected risk error, the generalization gap $G(\mathcal{X}, M^*)$ of DML is denoted as

$$G(\mathcal{X}, M^*) := L(\mathcal{X}, M^*) - L(X_{tr}, M^*) \tag{3}$$

The previous works [8, 18, 4, 2] attempt to study $G(\mathcal{X}, M^*)$ directly. To reach a tighter bound of $G(\mathcal{X}, M^*)$, these works study the impact of property of M^* and θ^* on $G(\mathcal{X}, M^*)$, proposing restraints or training protocols to adjust the property of optimal M^* *a priori* in the training phase. Orthogonal to these works, in this paper we investigate in the impact of *posterior* refinement operation on the metric matrix. Suppose that an optimal metric matrix M^* and corresponding θ^* have been obtained via certain optimization procedure, before evaluating the learned metric in the testing phase, we further refine the trained metric matrix $M' = g(M^*)$ by a matrix refinement operation $g : \mathcal{M} \to \mathcal{M}$. M' is assumed to be still symmetric positive semi-definite, which is the prerequisite that one square matrix can serve as a metric matrix. Then, our objective generalization gap $G(\mathcal{X}, M', M*)$ is represented as

$$G(\mathcal{X}, M', M^*) := L(\mathcal{X}, M') - L(X_{tr}, M^*) \tag{4}$$

which is the difference between the expected risk error under the metric of M' and the empirical error under the metric of M^*. Since by definition, posterior refinement will not consider the improvement of θ^* or M^*, θ^* and M^* can be regarded as constant variables, thus $L(X_{tr}, M^*)$ can be regarded as one definite and constant number. Then the upper bound of $G(\mathcal{X}, M', M^*)$ is purely determined by $L(\mathcal{X}, M')$. In the next section, we will analyze the upper bound of $L(\mathcal{X}, M')$, which has only one constant difference to the generalization gap.

3 Upper Bound of the Generalization Gap

In this section, we will derive the formula of the expected risk error under the posterior refined metric matrix $L(\mathcal{X}, M')$, and establish the link from its upper bound to the property of the refined metric matrix. Finally, based on our theorem, we will point out two principles to reduce the upper bound of $L(\mathcal{X}, M')$, which also lowers the upper bound of the generalization gap.

3.1 Formula of Expected Risk Error

Consider the general evaluation pipeline to evaluate a trained metric M on distribution $\mathcal{X}$. First, n instances are newly sampled to form the reference set denoted as D_{ref}. None of the instances in D_{ref} has been used for training. Let $X_{ref} = [x_1^{ref}, x_2^{ref}, \ldots, x_n^{ref}]$ denote the embedding matrix of D_{ref} generated by feature extractor function f. In practice, the testing set can be regarded as the reference set. Then, let $x_i{}' \in \mathcal{X}$ denote a random variable which is supposed to share the same category as x_i^{ref}. Let $X_i = [x_1^{ref}, x_2^{ref}, \ldots, x_{i-1}^{ref}, x_i{}', x_{i+1}^{ref}, \ldots, x_n^{ref}]$ denote the embedding matrix which is prepared to evaluate the error caused by $x_i{}'$ and let C_i denote the corresponding similarity sampling matrix. X_i only differs X_{ref} in the ith column. Then, the error on random variable $x_i{}'$ can be denoted as $L(X_i, M')$. Therefore, the expected risk error can be represented as the expectation of the error caused by all $x_i{}'$.

$$L(\mathcal{X}, M') = \mathop{\mathbb{E}}_{i}[L(X_i, M')] = \frac{1}{n}\sum_{i=1}^{n} L(X_i, M') \tag{5}$$

For $L(X_i, M')$, notice that X_i is a finite set with only one random variable $x_i{}'$. Then, $L(X_i, M')$ can be represented as the expectation over this random variable

$$L(X_i, M') = \frac{1}{P(M')|\mathcal{T}|}\int_{x_i{}' \in \mathcal{X}} p(x_i{}') tr(X_i^T M' X_i C_i) dx_i{}' \tag{6}$$

where $p(x_i{}')$ is the probability density function (PDF) of $x_i{}'$. Carrying Eq.6 into Eq.5, we get

$$L(\mathcal{X}, M') = \frac{1}{nP(M')|\mathcal{T}|}\sum_{i=1}^{n}\int_{x_i{}' \in \mathcal{X}} p(x_i{}') tr(X_i^T M' X_i C_i) dx_i{}' \tag{7}$$

Let $X' = [x_1{}', x_2{}', \ldots, x_n{}']$ denote the embedding matrix of n random variables. Since $x_i{}'$ is i.i.d, then $p(X') = p(x_1{}')p(x_2{}')\ldots p(x_n{}')$. Also notice that $\forall i, \int_{x_i{}' \in \mathcal{X}} p(x_i{}')dx_i{}' = 1$, thus we can always attach an additional integral on $x_j{}'$ such that the following equation holds

$$\int_{x_i{}' \in \mathcal{X}} p(x_i{}') tr(X_i^T M' X_i C_i) dx_i{}' = \int_{x_i{}' \in \mathcal{X}, x_j{}' \in \mathcal{X}} p(x_i{}')p(x_j{}') tr(X_i^T M' X_i C_i) dx_i{}' dx_j{}' \tag{8}$$

Then, Eq.7 can be further rewritten as

$$\begin{aligned} L(\mathcal{X}, M') &= \frac{1}{nP(M')|\mathcal{T}|}\sum_{i=1}^{n}\int_{x_1{}', \ldots, x_n{}' \in \mathcal{X}} p(x_1{}')\ldots p(x_n{}') tr(X_i^T M' X_i C_i) dx_1{}' \ldots dx_n{}' \\ &= \frac{1}{nP(M')|\mathcal{T}|}\sum_{i=1}^{n}\int_{X' \in \mathcal{X}^n} p(X') tr(X_i^T M' X_i C) dX' \\ &= \int_{X' \in \mathcal{X}^n} p(X')\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{P(M')|\mathcal{T}|} tr(X_i^T M' X_i C_i)\right) dX'. \end{aligned} \tag{9}$$

The last equality switch the order of $\frac{1}{n}\sum_{i=1}^{n}$ and $\int_{X'\in\mathcal{X}^n} p(X')$, because these two components are separable.

Eq. 9 points out the difference between the evaluation error on the testing set $L(X_{te}, M')$ and the expected error $L(\mathcal{X}, M')$. Consider the standard evaluation procedure of DML models. For each instance in the testing set, the rest of the testing set is regarded as the reference set. Then the set prepared to evluate the error of ith testing instance is $X_{te_i} = [x_1^{ref}, \ldots, x_{i-1}^{ref}, x_i^{te}, x_{i+1}^{ref}, \ldots x_n^{ref}] = X_{te} = X_{ref}$, here by assumption the testing set is used as the reference set. Thus the error caused by the ith testing instance is $L(X_{te_i}, M') = \frac{1}{P(M')|\mathcal{T}|} tr(X_{ref}^T M' X_{ref} C)$. The evaluation error on the testing set is the expected error across all testing instances, denoted as $L(X_{te}, M') = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{P(M')|\mathcal{T}|} tr(X_{ref}^T M' X_{ref} C)\right) = \frac{1}{P(M')|\mathcal{T}|} tr(X_{ref}^T M' X_{ref} C)$. Therefore, from Eq. 9, we known that actually, there is a gap between $L(X_{te}, M')$ and $L(\mathcal{X}, M')$, where $L(\mathcal{X}, M')$ further requires an integral over X'. Such a gap is caused by the oversight of the current evaluation procedure that, when evaluating the error of x_i^{te}, it should be regarded as a random variable instead of one fixed variable. Therefore, to give a more fundamental theoretical analysis of the generalization gap, we will analyze and try to reduce the upper bound of $L(\mathcal{X}, M')$ instead of $L(X_{te}, M')$.

For the upper bound of $L(\mathcal{X}, M')$, we only need to analysis the upper bound of $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{P(M')|\mathcal{T}|} tr(X_i^T M' X_i C_i)\right)$, since in Eq. 9 (1) $p(X')$ is determined by the intrinsic distribution of image space and the model parameters θ^*, thus changing the property of M' will not influence $p(X')$. So we neglect the impact of $p(X')$ on the upper bound of $L(\mathcal{X}, M')$. (2) $p(X')$ is a non-negative function, so the upper bound of $L(\mathcal{X}, M')$ is positively correlated to the upper bound of $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{P(M')|\mathcal{T}|} tr(X_i^T M' X_i C_i)\right)$. Then, in the following the upper bound of $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{P(M')|\mathcal{T}|} tr(X_i^T M' X_i C_i)\right)$ will be linked to the property of M'.

3.2 Upper Bound of Expected Error

In the following the uniform upper bound u of $tr(X_i^T M' X_i C_i)/P(M')$ across the index i will be derived, denoted as $tr(X_i^T M' X_i C_i)/P(M') \leq u$. Then the objective formula can be upper bounded by $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{P(M')|\mathcal{T}|} tr(X_i^T M' X_i C_i)\right) \leq \frac{u}{|\mathcal{T}|}$. Therefore, eliminating u can reach a tighter upper bound of expected risk error $L(\mathcal{X}, M')$. Suppose $|x_i' - x_i^{ref}|_2 \leq \epsilon_1$, where $|x|_2$ denote the L2 norm of vector x. Let $\|X\|_F$ denote the Frobenius norm of matrix X. First, consider the most simple and general case that $P(M') := \|M'\|_F$, then

Theorem 1. *$tr(X_i^T M' X_i C_i)/\|M'\|_F$ is upper bounded by*

$$tr({X_i}^T M' X_i C_i)/\|M'\|_F \leq \delta_1\delta_2 tr(M')/\|M'\|_F \tag{10}$$

where $\delta_1 := (\|X_{ref}\|_F + \epsilon_1)^2$, $\delta_2 := \sqrt{\frac{n}{m}(m-n)^2 + \frac{n^2-n}{m^2}(m-1)^2}$

Proof. See Supplementary A.2

Except $tr(M')/\|M'\|_F$, the dominating factors of other components, including $\epsilon_1, \delta_1, \delta_2$, are not related to the property of M'. Observing Eq. 10, it is easy to verify that reducing the trace of the learned metric matrix will result in a relative smaller $tr(M')/\|M'\|_F$. Then the first principle to guide the posterior refinement operation on the metric matrix can be summarized as

Principle 1 *If the refined matrix is symmetric positive semi-definite, then reducing the trace of the learned metric matrix leads to a tighter upper bound for expected risk error under the metric of the refined matrix.*

Theorem 1 points out the correlation of the generalization ability and the trace of the refined metric matrix. Since $tr(M')$ can also eliminate the influence of matrix scalar, in the following, we attempt to derive another upper bound of $tr(X_i^T M' X_i C_i)/P(M')$ when $P(M') := tr(M')$. The following theorem holds

Theorem 2. *$tr(X_i^T M' X_i C_i)/tr(M')$ is upper bounded by*

$$tr(X_i^T M' X_i C_i)/tr(M') \leq \delta_3 \|M'\|_F / tr(M') \tag{11}$$

where $\delta_3 = \epsilon_1 \sqrt{\delta_1} \delta_2 + \epsilon_1^2 \delta_2 + \delta_2 \|X_{ref}\|_F^2$.

Proof. See supplementary A.3.

The Theorem 2 is simple and straightforward. Since δ_3 is not dominated by the metric matrix anymore, it is easy to draw the following second principle

Principle 2 *If the refined matrix is symmetric, then fixing the trace and reducing the Frobenius norm of the learned metric matrix leads to a tighter upper bound for expected risk error under the metric of the refined matrix.*

4 Experiments

4.1 Experiment Setup

Dataset We conduct experiments on three widely-used benchmark datasets for deep metric learning: CUB-200-2011 (CUB) [28], Cars196 (Cars) [12] and Stanford Online Products (SOP) [16]. For CUB, we use the first half split of 100 classes with $5,864$ images for training and $5,924$ images from the last half split for testing. Similarly, for Cars, $8,054$ images of the first 98 classes are used for training and $8,131$ images of the rest classes are used for testing. For SOP, we follow the official dataset split using the first $11,318$ classes with $59,551$ images for training and the rest $11,316$ classes with $60,502$ instances for testing. Therefore, for the training set and testing set, neither the total instance number nor the class number differs much, supporting our assumption about DML datasets in Section 2.1.

Trained Models All the evaluated models have already been trained under the constraint of a certain loss function following the training protocols on powerful-benchmarker [15], a fair comparison platform for deep metric learning. To be more specific, BN-Inception [10] pretrained on ImageNet dataset [3] is adopted as the backbone model with output embedding dimension of 1024. A Multi-layer Perceptron (MLP) as the neck model further reduces the embedding dimension to 128. All mini-batches are constructed by arbitrarily sampling 32 (or 8) classes and 1 (or 4) instances per class for training data, resulting in a mini-batch size of 32. As for image augmentation, all images are resized into 256×256 pixels and then randomly cropped to patches of 227×227 pixels during training. RMSprop [22] under a fixed learning rate of 10^{-6} is employed to train the backbone model and the neck model simultaneously. The training is terminated when no improvement is gained on the validation set. 50 iterations of Bayesian optimization are run to find the best hyperparameters of the DML methods compared in [15]. Each iteration consists of 4-fold cross-validation. Therefore, the best performance yielded by the Bayesian optimization measures the upper bound of the discrimination ability that the corresponding DML method can achieve. We refer the readers to [15] for more details about the training protocol.

Evaluation Protocol Each run of the experiment consists of 4-fold cross-validation, generating 4 different packages of model, each model package contains the checkpoints of one backbone model and one neck model. For each fold, the accuracy of the corresponding model package is obtained, and therefore we can obtain 4 different accuracies for each DML method. Then the average of these 4 accuracies is reported as the final accuracy for this run.

For each DML method, the hyperparameters and the corresponding model checkpoints for the highest-accuracy run are provided by [15]. We directly fetch these checkpoints and follow the above-mentioned evaluation protocol to compare the testing accuracy under different posterior refinement methods.

4.2 Posterior Refinement Methods

From the view of deep metric learning, the backbone model serves as the feature extractor, mapping an input sample I_i to an embedding vector $x_i \in R^d$. Let $W \in R^{d' \times d}$ denote the weight of the neck model which conducts linear dimension reduction from d to d'. The similarity of two samples (I_i, I_j) can be represented as the inner product of two d' dimensional embeddings, denoted as $x_i^T(W^TW)x_j$. Therefore, the symmetric positive semi-definite matrix $M^* := W^TW$ can serve as the metric matrix, which is trainable as model parameters during the training procedure. To verify the correctness of our principles, we implement several refinement operations on the trained metric matrix M^*, namely

1. **Diagonal Restraint** which randomly restrains the diagonal elements of M^* to be zero. To be more specific, given a hyperparameter $0 \leq r \leq 1$, we randomly set one non-zero diagonal element of M^* to be zero and repeated

this operation until $tr(M') \leq (1-r)tr(M^*)$. In the extreme case when $r = 1$, all the diagonal elements of M^* are set to zero. Diagonal restraint is prepared to evaluate the correctness of Principle 1.

2. **Random Restraint** which randomly restrains the non-diagonal elements of M^* to be zero. Analogous to diagonal restraint, we randomly select an index pair $\{i, j\}(i \neq j)$, and set two elements to be zeros $m_{ij} = 0$, $m_{ji} = 0$. This operation is repeated until $\|{M_{off}}'\|_F \leq (1-r)\|M^*_{off}\|_F$, where M_{off} denote the off-diagonal part of the matrix. Random restraint is prepared to evaluate the correctness of Principle 2. To decouple the effect of matrix trace, we avoid setting the diagonal elements to zero, so that the accuracy diversification can be purely traced to $\|M'\|_F$.
3. **Identity Refinement** which set M' to be a $d \times d$ identity matrix regardless of M^*.

The above three restraint methods can satisfy the precondition in two principles as discussed in Section B.2, thus they can be used to verify the correctness of two principles. For diagonal restraint and random restraint, the hyperparameter r controls the degree to which the matrix restraint is carried out. A relative larger r requests that we have to restrain more elements of M^* to be zero, resulting in smaller $tr(M')$ for diagonal restraint, or smaller $\|M'\|_F$ for random restraint.

After the refinement operation, on the testing set we collect the embedding vectors generated by the trunk model, compute the pairwise similarity under the refined metric matrix, and compute the accuracy via a KNN classifier just as the standard evaluation pipeline.

Table 1: Comparison between the trained metric, identity matrix (marked as *) and the best refined matrix under random restraint or diagonal restraint (marked as †). The superscript $D/R, r$ represents such best refinement method is **D**iagonal or **R**andom restraint with restraint degree r. The best and the second best metric for each DML method is highlighted in red and blue, respectively.

Methods	CUB			Cars			SOP		
	MAP	Pre@1	RP	MAP	Pre@1	RP	MAP	Pre@1	RP
ArcFace[23]	21.51	60.12	32.39	17.48	72.88	27.66	42.23	71.98	45.15
ArcFace*	24.20	65.06	35.05	18.10	78.79	27.78	42.18	72.28	44.93
ArcFace†	24.18	64.90	$35.03^{R,1}$	18.66	78.80	$28.55^{D,1}$	43.08	73.02	$45.88^{D,1}$
ProxyNCA[14]	19.39	56.99	30.18	18.85	73.91	29.43	41.90	71.51	44.86
ProxyNCA*	22.16	63.64	33.03	18.43	80.10	28.27	41.96	71.79	44.75
ProxyNCA†	22.16	63.55	$33.03^{R,1}$	19.58	79.82	$29.76^{D,1}$	42.80	72.48	$45.65^{D,1}$
Triplet[27]	18.29	54.95	29.09	15.35	64.92	26.14	38.50	67.96	41.75
Triplet*	21.16	61.52	32.07	16.90	74.86	27.21	40.33	70.49	43.33
Triplet†	21.16	61.61	$32.08^{R,1}$	17.16	72.53	$27.82^{D,1}$	40.45	70.28	$43.55^{D,1}$

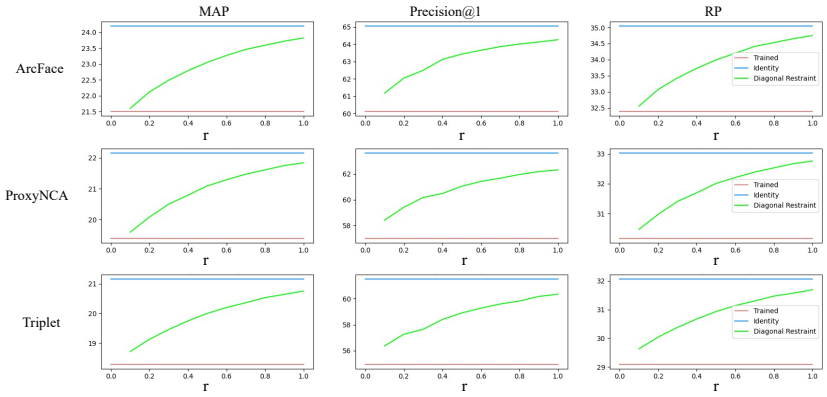

Fig. 2: Ablation of the restraint degree r for diagonal restraint on three DML methods. From left to right column enumerates the accuracy of MAP, Precision@1 and RP. Models are trained by ArcFace, ProxyNCA and Triplet.

4.3 Problem Observation

Recall that the motivation of this paper is founded on one observation that the identity matrix generalizes better than the trained metric matrix. To reveal this observation more comprehensively and specifically, in Table 1 we compare the performance of the trained metric, the identity matrix and the best performance of diagonal restraint or random restraint based metric matrix among three DML methods over three benchmark datasets. Of the three methods, ArcFace [23] is the SOTA method on CUB, ProxyNCA is a classical classification-based method, and Triplet loss is one classical pair-based method. Please refer to supplementary for full comparison across all methods in [15]. As can be clearly observed, for each DML method, the performance of identity matrix or restrained based metric has a remarkable margin in advance of the trained metric. Roughly speaking, the refined matrices improves MAP by 3%, precision@1 by 5%, RP by 3% on CUB, MAP by 1%, precision@1 by 7%, RP by 0.5% on Cars, and MAP by 1%, precision@1 by 2%, RP by 1% on SOP. Since identity refinement can be regarded as one special case of posterior refinement operation, we claim that there exist lightweight posterior refinement operations which can boost the performance of the trained metric significantly.

4.4 Verification of two Principles

Principle 1 To evaluate the correctness of principle 1, we conduct ablation study about the restraint degree r for diagonal restraint. A relatively larger r means that we have to set more diagonal elements of M' to be zero, leading to a relatively smaller $tr(M')/\|M'\|_F$. According to Principle 1, this will result in a tighter upper bound for the expected risk error, thus increasing the accuracy in practice. Therefore, we conduct diagonal restraint under different restraint

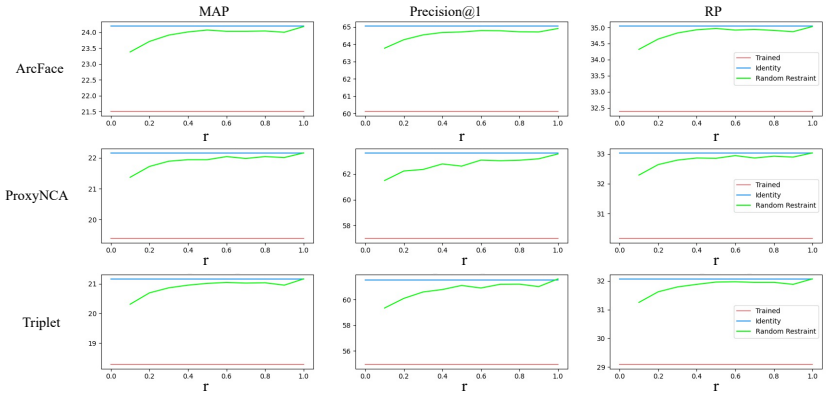

Fig. 3: Ablation of the restraint degree r for random restraint on three DML methods. From left to right column enumerates the accuracy of MAP, Precision@1 and RP. Models are trained by ArcFace, ProxyNCA and Triplet.

degrees r on the metric matrix trained by ArcFace, ProxyNCA and Triplet on CUB, and then compute the accuracy under the metric of the refined metric. As depicted in Fig. 2, better performance is achieved as r increases on three datasets consistently, which exactly follows the tendency in Principle 1.

Principle 2 A relatively larger restraint degree r for random restraint leads to a smaller Frobenius norm of M' while $tr(M')$ is fixed. According to Principle 2, this will lead to a tighter upper bound of the generalization gap, thus reflected as advanced performance. As illustrated in Fig. 3, all the DML methods benefit from a relative larger r. Also, notice that in the extreme case of $r = 1$ when all off-diagonal elements are all restrained to zero, random restraint converges to an identity matrix. Such observation implies the reason that the identity matrix generalizes better than the trained matrix because it has a reduced Frobenius norm. In the next we will give a more concrete discussion on the identity matrix.

4.5 Discussion of Identity Matrix

Based on the theoretical analysis and experimental results above, we attempt to explain the most original question that why an identity matrix generalizes better than the trained metric. Let M^* denote the metric matrix trained by ArcFace on CUB, and I denote an identity matrix having the same shape as M^*. In practice, $tr(M^*) = 42.72$ and $\|M^*\|_F^2 = 16.03$. In the following, we will compare the property of M^* and aI, where $a > 0$ is a scalar to preserve several properties consistent between M^* and aI, so that we can decouple the impact of two objective factors (trace and Frobenius norm) discovered in this paper. A numerical analysis to verify whether aI follows two principles is given as

1. **Principle 1** To control $\|aI\|_F = \|M^*\|_F$, then we have $a = \frac{\|M^*\|_F}{\sqrt{1024}}$. Thus the trace of $tr(aI) = \sqrt{1024}\|M^*\|_F > tr(M^*)$, which means that the identity matrix actually violates Principle 1.
2. **Principle 2** To control $tr(aI) = tr(M^*)$, then we have $a = \frac{tr(M^*)}{1024}$. Thus the Frobenius norm of $\|aI\|_F = \frac{tr(M^*)}{32} < \|M^*\|_F$. Therefore, identity matrix follows Principle 2. Such conclusion can also be confirmed by such observation that, for random restraint when $r = 1$, all off-diagonal elements are restrained to zero, then the refined matrix approximates an identity matrix. Correspondingly, as depicted in Fig. 2 random restraint will converge to identity matrix as restraint degree increases.

Therefore, one conclusion can be drawn that an identity matrix generalizes better than a trained metric because it has a limited Frobenius norm. It is worth noticing that one good posterior refinement operation does not have to follow two principles at the same time. As long as one principle is followed, a relatively tighter upper bound of the generalization gap can be established.

5 Related Work

5.1 Classical DML Methods

Deep metric learning attempts to learn a powerful embedding space where instances can be discriminated based on the inter-instance similarity. Early metric learning losses, such as contrastive loss [6] and triplet loss [27], construct instance pairs from the training set, maximizing the similarity of positive pairs while minimizing that of negative ones. These methods are known as *pair-based method*. Some of the recent pair-based methods attempt to consider adaptive margin for flexible expected gap between the similarity of positive pairs and negative pairs [24, 29], whereas others construct more data pairs to consider denser relationships among data [16, 19]. At the same time, some studies [26, 20] provide efficient approaches for collecting and weighting informative pairs based on gradient analysis. Meanwhile, other methods leverage concepts defined in signal processing to devise new similarity functions [31, 1]. However, pair-based methods are known to suffer from the slow convergence speed, which is caused by the manner of learning on enormous but low informative data pairs [29].

In contrast to pair-based methods, another mainstream of studies, called *proxy-based method*, resolve such problem by introducing learnable category representations (called proxies) into the organization of the loss formula. These methods only consider data-to-proxy relations, which significantly reduces the computation burden since the number of objective proxy-data pairs is much less than data pairs. ProxyNCA [14] is one of the first studies to introduce this paradigm. ProxyNCA++ [21] and proxy anchor loss [11] integrate several improvements into Proxynca, including backbone enhancement, carefully designed learning rate scheduler, temperature scaling and denser proxy-to-data relations.

Moreover, GS-TRS loss [5] and softtriple [17] loss attempt to construct multiple-center embedding structure such that intra-class variance and inter-class diversity can be handled separately.

5.2 Generalization Bound of DML

Despite the remarkable success the popular DML methods have achieved, there are no theoretical guarantees that these methods have strong generalization ability, which refers to the capacity that the trained metric can yield approximate performance on unseen data. Recently, a few works attempt to theoretically analyze the generalization ability of DML methods and propose several principles to reduce the upper bound of the generalization gap. Some of these works propose regularizers on the learned metric. Cao *et al.* [2] first proposes to learn a sparse metric via L1 norm regularization. Roth *et al.* [18] conduct comprehensive experiments to show that ranking-based DML methods are hurt by the excessive compressed level of instance representation, thus proposing a simple technique to regularize the compression of the learned embedding space. Others consider the impact of several important components of the backbone model. Huai *et al.* [8] involve dropout into the generalization bound, thus proving the benefit of taking an adaptive dropout into account during the training process. Further, Dong *et al.* [4] suggests that early stopping as well as the Lipschitz smooth loss function and classifier has a positive influence on the generalization error.

The existing DML generalization theorems all assume that the trained metric will be directly employed in evaluation. Therefore, they propose methods to introduce the discovered prior knowledge about the generalization into the training procedure to enhance the current DML pipeline. Orthogonality to these methods, in this paper we investigate in the impact of the posterior refinement on the metric matrix, which is a lightweight data-free operation but also significantly improves the generalization ability of the trained metric.

6 Conclusion

In this paper, we uncover an observation that, before the testing phase, a lightweight posterior refinement operation on the learned metric can significantly improve the generalization ability of the deep metric learning models. Based on the theoretical analysis of the generalization bound the refined matrix, we propose two simple principles to guide the refinement operations, and conduct experiments on three benchmark datasets to verify the correctness of these principles. The theories and experiments prove that posterior refinement can serve as a lightweight plug-in to boost the performance of DML models dramatically.

Acknowledgements

Yi Xu is supported in part by the National Natural Science Foundation of China (62171282, 111 project BP0719010, STCSM 18DZ2270700), the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

References

1. Cakir, F., He, K., Xia, X., Kulis, B., Sclaroff, S.: Deep metric learning to rank. In: CVPR. pp. 1861–1870 (2019)
2. Cao, Q., Guo, Z.C., Ying, Y.: Generalization bounds for metric and similarity learning. Machine Learning **102**(1), 115–132 (2016)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
4. Dong, M., Yang, X., Zhu, R., Wang, Y., Xue, J.H.: Generalization bound of gradient descent for non-convex metric learning. Advances in Neural Information Processing Systems **33**, 9794–9805 (2020)
5. Em, Y., Gag, F., Lou, Y., Wang, S., Huang, T., Duan, L.Y.: Incorporating intra-class variance to fine-grained visual recognition. In: 2017 IEEE International Conference on Multimedia and Expo (ICME). pp. 1452–1457. IEEE (2017)
6. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR. vol. 2, pp. 1735–1742. IEEE (2006)
7. Harwood, B., Kumar BG, V., Carneiro, G., Reid, I., Drummond, T.: Smart mining for deep metric learning. In: ICCV. pp. 2821–2829 (2017)
8. Huai, M., Xue, H., Miao, C., Yao, L., Su, L., Chen, C., Zhang, A.: Deep metric learning: The generalization analysis and an adaptive algorithm. In: IJCAI. pp. 2535–2541 (2019)
9. Huang, C., Loy, C.C., Tang, X.: Local similarity-aware deep feature embedding. arXiv preprint arXiv:1610.08904 (2016)
10. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
11. Kim, S., Kim, D., Cho, M., Kwak, S.: Proxy anchor loss for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3238–3247 (2020)
12. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: ICCV workshops. pp. 554–561 (2013)
13. Liu, K., Bellet, A.: Escaping the curse of dimensionality in similarity learning: Efficient frank-wolfe algorithm and generalization bounds. Neurocomputing **333**, 185–199 (2019)
14. Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S.: No fuss distance metric learning using proxies. In: ICCV. pp. 360–368 (2017)
15. Musgrave, K., Belongie, S., Lim, S.N.: A metric learning reality check. arXiv preprint arXiv:2003.08505 (2020)
16. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: CVPR. pp. 4004–4012 (2016)
17. Qian, Q., Shang, L., Sun, B., Hu, J., Li, H., Jin, R.: Softtriple loss: Deep metric learning without triplet sampling. In: ICCV. pp. 6450–6458 (2019)
18. Roth, K., Milbich, T., Sinha, S., Gupta, P., Ommer, B., Cohen, J.P.: Revisiting training strategies and generalization performance in deep metric learning. In: International Conference on Machine Learning. pp. 8242–8252. PMLR (2020)
19. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. NIPS **29**, 1857–1865 (2016)
20. Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., Wei, Y.: Circle loss: A unified perspective of pair similarity optimization. In: CVPR. pp. 6398–6407 (2020)

21. Teh, E.W., DeVries, T., Taylor, G.W.: Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. arXiv preprint arXiv:2004.01113 (2020)
22. Tieleman, T., Hinton, G.: Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning (2012)
23. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE Signal Processing Letters **25**(7), 926–930 (2018)
24. Wang, J., Zhou, F., Wen, S., Liu, X., Lin, Y.: Deep metric learning with angular loss. In: ICCV. pp. 2593–2601 (2017)
25. Wang, X., Hua, Y., Kodirov, E., Hu, G., Robertson, N.M.: Deep metric learning by online soft mining and class-aware attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5361–5368 (2019)
26. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. In: CVPR. pp. 5022–5030 (2019)
27. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: NIPS. pp. 1473–1480 (2006)
28. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)
29. Wu, C.Y., Manmatha, R., Smola, A.J., Krahenbuhl, P.: Sampling matters in deep embedding learning. In: ICCV. pp. 2840–2848 (2017)
30. Yang, F., Wang, Z., Xiao, J., Satoh, S.: Mining on heterogeneous manifolds for zero-shot cross-modal image retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 12589–12596 (2020)
31. Yuan, T., Deng, W., Tang, J., Tang, Y., Chen, B.: Signal-to-noise ratio: A robust distance metric for deep metric learning. In: CVPR. pp. 4815–4824 (2019)
32. Zhu, Y., Yang, M., Deng, C., Liu, W.: Fewer is more: A deep graph metric learning perspective using fewer proxies. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 17792–17803. Curran Associates, Inc. (2020)