Computer Vision and Pattern Recognition (CVPR), June 2019

Re-ranking via Metric Fusion for Object Retrieval and Person Re-identification

Song Bai[1] Peng Tang[2] Philip H.S. Torr[1] Longin Jan Latecki[3]
[1]University of Oxford [2]Huazhong University of Science and Technology [3]Temple University
{songbai.site,tangpeng723}@gmail.com, philip.torr@eng.ox.ac.uk, latecki@temple.edu

Abstract

This work studies the unsupervised re-ranking procedure for object retrieval and person re-identification with a specific concentration on an ensemble of multiple metrics (or similarities). While the re-ranking step is involved by running a diffusion process on the underlying data manifolds, the fusion step can leverage the complementarity of multiple metrics.

We give a comprehensive summary of existing fusion with diffusion strategies, and systematically analyze their pros and cons. Based on the analysis, we propose a unified yet robust algorithm which inherits their advantages and discards their disadvantages. Hence, we call it Unified Ensemble Diffusion (UED). More interestingly, we derive that the inherited properties indeed stem from a theoretical framework, where the relevant works can be elegantly summarized as special cases of UED by imposing additional constraints on the objective function and varying the solver of similarity propagation. Extensive experiments with 3D shape retrieval, image retrieval and person re-identification demonstrate that the proposed framework outperforms the state of the arts, and at the same time suggest that re-ranking via metric fusion is a promising tool to further improve the retrieval performance of existing algorithms.

1. Introduction

Due to the advance in the acquisition, storage, and sharing of visual content, the image and multimedia collections have shown a continuous and consistent growth, both in scope and diversity. Consequently, the development of methods for indexing and retrieving such information has become essential. Given a query instance, the goal of visual retrieval is to find objects sharing similar visual appearances with the query in a large database. Therefore, a reliable metric (or similarity) function is vital to the retrieval performance.

However, traditional object retrieval systems perform only pairwise comparisons, *i.e.*, computing distance (or similarity) measures between object pairs and ignoring the contextual information encoded in the relationships among objects. To address this issue, re-ranking approaches [34, 4, 5, 22] have been proposed for the sake of refining the retrieval results without the need of user intervention. Such methods (*e.g.*, manifold ranking [72], diffusion process [11]) replace the pairwise distances by more global similarity measures, capable of analyzing data collections more globally and taking into account the underlying manifold structure to reveal the intrinsic relationship between objects.

Meanwhile, with the long-standing development of feature learning, plenty of visual descriptors have been proposed, from the conventional handcrafted ones [57, 56, 46, 30] to deep-learned ones [12, 61, 67, 26]. Different visual descriptors generally focus on different visual characteristics of objects. As a result, significant efforts [37] have been devoted recently to metric fusion to leverage the complementary nature. Generally, metric (or similarity) fusion can be done in any stage of a typical retrieval pipeline (*e.g.*, feature learning stage [38], indexing stage [49, 66, 40]). In this work, we consider metric fusion in the re-ranking stage, particularly diffusion process [11], to capture the geometrical structure of multiple data manifolds.

Existing fusion with diffusion methods can be coarsely divided into three categories. Naive Fusion (NF) simply averages the edge weights of multiple affinity graphs, such as locally constrained mixed diffusion [33], graph fusion [69, 68], and Yang *et al*. [62]. In order to combine two distinct and complementary metrics, Tensor Product Fusion (TPF) [73] considers a homogeneous fusion on a tensor product graph. To handle noisy input metric, Regularized Ensemble Diffusion (RED) [8] performs similarity learning and weight learning simultaneously to maximize the smoothness of multiple graph-based manifolds.

As detailed in Sec. 2, NF is the fastest among these methods, but it is extremely susceptible to noisy similarities. By contrast, TPF considers the interplay of two similarities, attaining robustness to noise to a certain extent. However, it can only fuse two similarities each time. Although RED can eliminate the influence of noises via a dynamic weight

learning mechanism, it is relatively computationally expensive as the diffusion step must be done for each input individually.

With these observations, we propose in this work a new fusion with diffusion algorithm called Unified Ensemble Diffusion (UED). The primary contributions of UED are three folds:

1) UED combines the advantages of three existing types of fusion with diffusion methods without inheriting their drawbacks. In particular, UED is more robust to noisy input than RED, since it considers the interplay of two similarities as TPF does. Meanwhile, it can handle more than two similarities, instead of merely two in the case of TPF. Furthermore, the diffusion step of UED can be executed much faster than RED, almost as fast as naive fusion. We will demonstrate those properties both theoretically (see Sec. 3) and experimentally (see Sec. B).

2) More importantly, by deeply analyzing the relationship between UED and existing methods, we observe that the inherited properties indeed stem from a unified framework, where all those methods can be summarized as special cases of UED. The inherent differences lie in the additional constraints on the objective function and the variation of similarity propagation (see Sec. 4).

3) UED has undergone a careful design of formulation and derivation. Unfortunately, it becomes a non-convex optimization, which is hard to solve. A by-product contribution of our work is, for the first time to our knowledge, to introduce the replicator equation [41, 42] as a powerful optimizer to learn the metric weights in the re-ranking stage.

Extensive experiments are conducted with 3D shape retrieval on the ModelNet40 [60] and ModelNet10 datasets, image retrieval on the Holidays [23] and Ukbench [35] datasets, and person re-identification on the Market-1501 dataset [70]. The state-of-the-art performance firmly demonstrates the effectiveness of the proposed framework.

2. Metric Fusion Revisited

Let $\mathcal{G} = \{\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^M\}$ be a multi-graph, where $\mathcal{G}^\mu = (\mathbf{X}, \mathbf{W}^\mu)$ is the μ-th ($1 \leq \mu \leq M$) affinity graph parameterized by the μ-th metric (or similarity). The vertex set $\mathbf{X} = \{x_1, x_2, \ldots, x_N\}$ denotes the objects and $\mathbf{W}^\mu \in \mathbb{R}^{N\times N}$ denotes the adjacency matrix with W^μ_{ij} being the initial similarity between x_i and x_j associated with the μ-th metric. Usually, a transition matrix is defined via $\mathbf{S}^\mu = (\mathbf{D}^\mu)^{-1/2}\mathbf{W}^\mu(\mathbf{D}^\mu)^{-1/2}$, where $\mathbf{D}^\mu \in \mathbb{R}^{N\times N}$ is a diagonal matrix with elements $D^\mu_{ii} = \sum_{j=1}^N W^\mu_{ij}$. The basic objective is to learn a new similarity $\mathbf{A} \in \mathbb{R}^{N\times N}$ on $\mathcal{G}$ in an unsupervised manner so that the indexed candidate images for a given query (or probe) can be re-ranked.

To enable re-ranking, various methodologies can be used, such as learning to rank [9], metric learning [37], manifold ranking [72], *etc*. In this work, we consider a representative branch called diffusion process [11] in retrieval, upon which we build the fusion paradigm to integrate multiple metrics. Among the variants of diffusion process summarized in [11], we select tensor product diffusion as the backbone as it has been demonstrated [63] to be more robust in the scope of object retrieval.

2.1. Naive Fusion

Naive Fusion (NF) is a two-step solution:

Fusion Step. Simply average the multiple similarities to generate the transition matrix as

$$\mathbf{S} = \frac{1}{M}\sum_{\mu=1}^{M}\mathbf{S}^\mu. \tag{1}$$

Diffusion Step. Run a diffusion process with $\mathbf{S}$ to obtain the target similarity $\mathbf{A}$ as

$$\mathbf{A}^{(t+1)} = \alpha\mathbf{S}\mathbf{A}^{(t)}\mathbf{S}^{\mathrm{T}} + (1-\alpha)\mathbf{I}, \tag{2}$$

where t is the number of iteration, $\alpha \in (0, 1)$ is a trade-off parameter, and $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix. As the transition matrix $\mathbf{S}$ is a symmetric matrix, we will interchangeably use $\mathbf{S} = \mathbf{S}^{\mathrm{T}}$ subsequently.

It is proven [4, 5] that after a sufficient number of iterations, Eq. (2) converges to

$$\mathbf{A}^* = (1-\alpha)\mathrm{vec}^{-1}\left((\mathbf{I} - \alpha\mathbf{S}\otimes\mathbf{S})^{-1}\mathrm{vec}(\mathbf{I})\right), \tag{3}$$

where $\otimes$ denotes the Kronecker product, $\mathrm{vec}(\cdot)$ is the vectorization of the input matrix by stacking its columns one by one, and its inverse function is vec^{-1}. To simplify the notation, we will use $\vec{\mathbf{Y}} = \mathrm{vec}(\mathbf{Y})$ for any input matrix $\mathbf{Y}$.

2.2. Tensor Product Fusion

Tensor Product Fusion (TPF) is a one-step solution:

Fusion with Diffusion Step. Simultaneously fuses two metrics in one diffusion step. When fusing the μ-th and the ν-th affinity graph, it is defined as

$$\mathbf{A}^{(t+1)} = \alpha\mathbf{S}^\nu\mathbf{A}^{(t)}\mathbf{S}^\mu + (1-\alpha)\mathbf{I}. \tag{4}$$

It is proven [73] that after a sufficient number of iterations, Eq. (4) converges to

$$\mathbf{A}^* = (1-\alpha)\mathrm{vec}^{-1}\left((\mathbf{I} - \alpha\mathbf{S}^\mu\otimes\mathbf{S}^\nu)^{-1}\vec{\mathbf{I}}\right). \tag{5}$$

2.3. Regularized Ensemble Diffusion

Regularized Ensemble Diffusion (RED) [8] is a two-step solution proposed recently:

Diffusion Step. Given $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_M\}$ with β_μ ($1 \leq \mu \leq M$) being the weight of the μ-th affinity graph, the diffusion step of RED is defined as

$$\mathbf{A}^{(t+1)} = \sum_{\mu=1}^{M} \alpha_\mu \mathbf{S}^\mu \mathbf{A}^{(t)} \mathbf{S}^\mu + (1 - \sum_{\mu=1}^{M} \alpha_\mu)\mathbf{I}, \quad (6)$$

where

$$\alpha_\mu = \frac{\beta_\mu}{\gamma + \sum_{\mu'=1}^{M} \beta_{\mu'}}. \quad (7)$$

Therein, $\gamma > 0$ is a small weight constant to ensure that the state of convergence

$$\mathbf{A}^* = \mathrm{vec}^{-1}\Big((1 - \sum_{\mu=1}^{M} \alpha_\mu)(\mathbf{I} - \sum_{\mu=1}^{M} \alpha_\mu \mathbf{S}^\mu \otimes \mathbf{S}^\mu)^{-1}\vec{\mathbf{I}}\Big) \quad (8)$$

can be obtained.

Fusion Step. The vector with metric weight $\boldsymbol{\beta}$ is not determined empirically. RED can dynamically learn the metric weights to amplify the contributions of discriminative affinity graphs and suppress those of noisy ones.

By initializing with equal weights $\frac{1}{M}$, the weight $\boldsymbol{\beta}$ can be optimized via coordinate descent. It has been proven that by alternating the diffusion step and the fusion step, an optimal similarity $\mathbf{A}^*$ and weight configuration $\boldsymbol{\beta}$ can be derived. Details can be found in [8].

2.4. Summary of Pros and Cons

The three existing types of fusion methods, including Naive Fusion (NF), Tensor Product Fusion (TPF), and Regularized Ensemble Diffusion (RED), have different pros and cons.

First, NF is the most efficient one. As can be seen from Eq. (1), NF conducts the fusion step of input similarities first, then the diffusion step is only executed once. However, it is quite vulnerable to noisy similarities as it weights each input equally. As a consequence, when less discriminative similarities exist, the retrieval performance of NF may easily deteriorate.

Second, TPF considers the complementarity and the interplay of two distinct similarities, as shown in Eq. (4). In comparison, NF and RED both consider input similarities individually, by simply averaging them with equal weights (see Eq. (1)) or dynamic weights (see Eq. (6)). However, one primary defect of TPF is that it can only tackle two inputs, limiting its promotion and usage where more than two metrics are available.

At last, among the three methods, RED is the most robust one to noisy similarities since it exerts a robust weight learning paradigm to the diffusion step. However, as Eq. (6) says, each diffusion step has to be done for each input similarity individually. Hence, it is more computationally expensive although the scale of time complexity is the same as NF and TPF. Interested readers can refer to [8] for more detailed analysis.

To address the limitations of existing types of fusion methods, we will present a novel method called Unified Ensemble Diffusion (UED) in Sec. 3 which inherits the advantages of those methods. More interestingly, we theoretically analyze in Sec. 4 that the inherited advantages stem from a unified framework, where NF, TPF, and RED can be elegantly summarized as special cases of UED.

3. Proposed Method

A pertinent suggestion of Unified Ensemble Diffusion (UED) is to first compute a weighted average of input similarities as

$$\mathbf{S} = \sum_{\mu=1}^{M} \beta_\mu \mathbf{S}^\mu, \quad (9)$$

where the weight $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_M\}$ will be learned afterwards. Although Eq. (9) appears to be a simple modification of naive fusion, we will demonstrate in this section that it leads to some nice mathematical properties and practical benefits (*e.g.*, it allows us to consider the interplay of all pairs of affinity graphs), which constitutes the base for the core contribution of this work in Sec. 4.

3.1. Objective Function

UED learns the target similarity $\mathbf{A}$ by solving the following optimization problem

$$\begin{aligned} &\min_{\mathbf{A},\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}} \mathbf{H} \boldsymbol{\beta} + \gamma \|\mathbf{A} - \mathbf{I}\|_{\mathrm{F}} + \eta \|\boldsymbol{\beta}\|_2^2, \\ &s.t.\ \boldsymbol{\beta} \in \Delta = \{\boldsymbol{\beta} \in \mathbb{R}^{M \times 1} : \boldsymbol{\beta} \geq 0,\ \|\boldsymbol{\beta}\|_1 = 1\}, \end{aligned} \quad (10)$$

where matrix $\mathbf{H} \in \mathbb{R}^{M \times M}$ with its entries defined as

$$\begin{aligned} H^{\mu\nu} &= \frac{1}{2} \sum_{i,j,k,l=1}^{N} W_{ij}^\mu W_{kl}^\nu \Big(\frac{A_{ki}}{\sqrt{D_{ii}^\mu D_{kk}^\nu}} - \frac{A_{lj}}{\sqrt{D_{jj}^\mu D_{ll}^\nu}}\Big)^2 \\ &= \vec{\mathbf{A}}^{\mathrm{T}} (\mathbf{I} - \mathbf{S}^\mu \otimes \mathbf{S}^\nu) \vec{\mathbf{A}} \end{aligned} \quad (11)$$

measures the smoothness of $\mathbf{A}$ with respect to all the input similarity pairs $\mathbf{W}^\mu$ ($1 \leq \mu \leq M$) and $\mathbf{W}^\nu$ ($1 \leq \nu \leq M$). $\|\mathbf{A} - \mathbf{I}\|_{\mathrm{F}}$ computes the difference of $\mathbf{A}$ from the identity matrix $\mathbf{I}$, meaning that the self-similarity should be preserved with the weight $\gamma > 0$. $\|\boldsymbol{\beta}\|_2^2$ computes the squared L_2 norm of $\boldsymbol{\beta}$, whose contribution to the overall loss is weighted by $\eta > 0$ to avoid overfitting to a specific input.

3.2. Derivation

As there are two variables to learn, *i.e.*, the target similarity $\mathbf{A}$ and the weight configuration $\boldsymbol{\beta}$, we decompose

Eq. (10) into two sub-problems, then adopt an alternating manner to solve the optimization problem.

Diffusion Step. When learning $\mathbf{A}$, we fix $\boldsymbol{\beta}$. Consequently, the third term in Eq. (10) is a constant and can be omitted. Then, Eq. (10) is equivalent to

$$\min_{\mathbf{A}} \sum_{\mu,\nu=1}^{M} \beta_\mu \beta_\nu \vec{\mathbf{A}}^{\mathrm{T}} (\mathbf{I} - \mathbf{S}^\mu \otimes \mathbf{S}^\nu) \vec{\mathbf{A}} + \gamma \|\vec{\mathbf{A}} - \vec{\mathbf{I}}\|_2^2. \quad (12)$$

By taking the partial derivative with respect to $\vec{\mathbf{A}}$, we obtain

$$2 \sum_{\mu,\nu=1}^{M} \beta_\mu \beta_\nu (\mathbf{I} - \mathbf{S}^\mu \otimes \mathbf{S}^\nu) \vec{\mathbf{A}} + 2\gamma(\vec{\mathbf{A}} - \vec{\mathbf{I}}). \quad (13)$$

By setting it to zero, we derive the closed-form solution

$$\vec{\mathbf{A}} = \frac{\gamma}{\Lambda} (\mathbf{I} - \frac{1}{\Lambda} \sum_{\mu,\nu=1}^{M} \beta_\mu \beta_\nu \mathbf{S}^\mu \otimes \mathbf{S}^\nu)^{-1} \vec{\mathbf{I}}, \quad (14)$$

where

$$\Lambda = \gamma + \sum_{\mu,\nu=1}^{M} \beta_\mu \beta_\nu = \gamma + 1. \quad (15)$$

By applying vec^{-1} to both sides of Eq. (14), the optimal solution $\mathbf{A}$ can be obtained.

To efficiently learn $\mathbf{A}$ in practice, we use an iteration-based solver given as

$$\mathbf{A}^{(t+1)} = \frac{1}{\Lambda} (\sum_{\nu=1}^{M} \beta_\nu \mathbf{S}^\nu) \mathbf{A}^{(t)} (\sum_{\mu=1}^{M} \beta_\mu \mathbf{S}^\mu) + \frac{\gamma}{\Lambda} \mathbf{I}. \quad (16)$$

By substituting Eq. (9) into Eq. (16), one can simplify it to

$$\mathbf{A}^{(t+1)} = \frac{1}{\Lambda} \mathbf{S} \mathbf{A}^{(t)} \mathbf{S} + \frac{\gamma}{\Lambda} \mathbf{I}. \quad (17)$$

A key observation drawn from Eq. (17) is that UED firstly computes a weighted average of multiple input similarities and conducts one diffusion step in one trial. Compared with NF (Eq. (2)), the diffusion step of UED is adequately efficient but less susceptible to noise owing to a weight learning mechanism. Compared with RED defined in Eq. (6) which needs to conduct a diffusion step for each input similarity individually, the diffusion step of UED is more computationally efficient because only one diffusion step is enough for multiple input similarities.

Now, we prove the iteration in Eq. (16) can approximate the closed-form solution in Eq. (14). Eq. (16) is equivalent to

$$\mathbf{A}^{(t+1)} = \frac{1}{\Lambda} \sum_{\mu,\nu=1}^{M} \beta_\nu \beta_\mu \mathbf{S}^\nu \mathbf{A}^{(t)} \mathbf{S}^\mu + \frac{\gamma}{\Lambda} \mathbf{I}. \quad (18)$$

By applying $\mathrm{vec}(\cdot)$ to its both sides and using the property of Kronecker product, we have

$$\vec{\mathbf{A}}^{(t+1)} = \frac{1}{\Lambda} \sum_{\mu,\nu=1}^{M} \beta_\mu \beta_\nu (\mathbf{S}^\mu \otimes \mathbf{S}^\nu) \vec{\mathbf{A}}^{(t)} + \frac{\gamma}{\Lambda} \vec{\mathbf{I}}. \quad (19)$$

As proven in the supplementary material, Eq. (19) converges to the closed-form solution in Eq. (14). To see this directly, one could set $\vec{\mathbf{A}}^{(t+1)} = \vec{\mathbf{A}}^{(t)}$ in Eq. (19). Then, the solution would look like Eq. (14).

Fusion Step. When learning $\boldsymbol{\beta}$, we fix $\mathbf{A}$. Consequently, the second term in Eq. (10) is a constant and can be omitted. Then, the objective function becomes

$$\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}} \mathbf{H} \boldsymbol{\beta} + \eta \|\boldsymbol{\beta}\|_2^2, \quad s.t.\ \boldsymbol{\beta} \in \Delta, \quad (20)$$

which is an optimization of a quadratic function on the simplex Δ. Unfortunately, Eq. (20) is not guaranteed to be a convex optimization with respect to $\boldsymbol{\beta}$, *e.g.*, $\mathbf{H} + \eta\mathbf{I}$ is not positive semi-definitive.

To address this issue, we prove that after some algebraic transformations, a replicator equation [41, 42] can be used to obtain a proper local maximizer of the following equivalent objective function

$$\max_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}} \bar{\mathbf{H}} \boldsymbol{\beta}, \quad s.t.\ \boldsymbol{\beta} \in \Delta, \quad (21)$$

where $\bar{\mathbf{H}} = -\mathbf{H}/2 - \mathbf{H}^{\mathrm{T}}/2 - \eta\mathbf{I} + \mathbf{C}$ and $\mathbf{C} \in \mathbb{R}^{M \times M}$ is a matrix with all its entries equal to the maximum element of $(\mathbf{H}/2 + \mathbf{H}^{\mathrm{T}}/2 + \eta\mathbf{I})$. Due to the space limitation, the detailed derivation is put in the supplementary material. Then, Eq. (21) can be solved by using the replicator equation as

$$\boldsymbol{\beta}^{(t+1)} = \frac{\boldsymbol{\beta}^{(t)} \odot \bar{\mathbf{H}} \boldsymbol{\beta}^{(t)}}{{\boldsymbol{\beta}^{(t)}}^{\mathrm{T}} \bar{\mathbf{H}} \boldsymbol{\beta}^{(t)}}, \quad (22)$$

where t is the number of iteration and $\odot$ denotes the element-wise multiplication. Two conditions need to be satisfied for the sake of the convergence of replicator equation [31]. First, $\bar{\mathbf{H}}$ is symmetric and all its entries are non-negative, which can be simply obtained from the definition of $\bar{\mathbf{H}}$. Second, every trajectory staring in the simplex Δ will remain in the simplex. To this end, we need to prove the L_1 norm of $\boldsymbol{\beta}^{(t+1)}$ is always equal to 1. Equivalently, we need to prove the L_1 norm of the numerator of Eq. (22) is equal to the denominator of Eq. (22). It holds, since

$$\begin{aligned} \|\boldsymbol{\beta}^{(t)} \odot \bar{\mathbf{H}} \boldsymbol{\beta}^{(t)}\|_1 &= \sum_{\mu=1}^{M} \beta_\mu^{(t)} \sum_{\nu=1}^{M} \bar{H}^{\mu\nu} \beta_\nu^{(t)} \\ &= \sum_{\mu,\nu=1}^{M} \beta_\mu^{(t)} \bar{H}^{\mu\nu} \beta_\nu^{(t)} = {\boldsymbol{\beta}^{(t)}}^{\mathrm{T}} \bar{\mathbf{H}} \boldsymbol{\beta}^{(t)}. \end{aligned} \quad (23)$$

Algorithm 1: Unified Ensemble Diffusion

Input:
M adjacency matrices $\{W^\mu\}_{\mu=1}^M \in \mathbb{R}^{N\times N}$, γ, η.
Output:
The target similarity $\mathbf{A}$.
begin
  Initialize the weight $\beta_\mu = \frac{1}{M}, \forall \mu$.
  repeat
    Compute $\mathbf{S}$ using Eq. (9).
    Update $\mathbf{A}$ using $\mathbf{S}$ and Eq. (17).
    Compute $\mathbf{H}$ using Eq. (11).
    Update $\boldsymbol{\beta}$ using $\mathbf{H}$ and Eq. (22).
  until *convergence*
  return $\mathbf{A}$

We alternate the diffusion step and the fusion step. The whole optimization is guaranteed to converge to an equilibrium. The overall procedure is summarized in Alg. 1. Comparing with the previous works, UED possesses some nice properties, as we will state in Sec. 4.

4. A Unified Framework

In this section, we demonstrate that existing fusion methods can be summarized in a unified framework defined by the proposed Unified Ensemble Diffusion (UED).

4.1. Regularization on Simplex

Recall the objective function of UED in Eq. (10), and a unified framework can be built by imposing an additional simplex Δ_o. Then, the constraint becomes

$$\boldsymbol{\beta} \in \Delta \cap \Delta_o, \tag{24}$$

which is the intersection of the original simplex Δ of UED and the additional simplex Δ_o.

Naive Fusion sets Δ_o to

$$\Delta_o = \{\boldsymbol{\beta} : \beta_\mu = \frac{1}{M},\ \forall \mu\}, \tag{25}$$

which means that all input similarities have equal weights and keep unchanged.

Tensor Product Fusion sets Δ_o to

$$\Delta_o = \{\boldsymbol{\beta} :\ \textit{if}\ \mu = \nu,\ \beta_\mu = \beta_\nu = 0;\ \textit{else}\ = 1\} \tag{26}$$

which means that only two different similarities are fused, both having weight 1.

Regularized Ensemble Diffusion sets Δ_o to

$$\Delta_o = \{\boldsymbol{\beta} : \beta_\mu\beta_\nu = 0,\ \forall \mu \neq \nu\}, \tag{27}$$

which means that no interplay between two different similarities are encouraged. All the input similarities are fused individually.

4.2. Variation of Iteration

Different regularizations on the simplex Δ_o are subjected to different iteration-based solver. Recall the iteration-based solver of UED in Eq. (16) and Eq. (17). Then, a unified framework can be built as follows.

Naive Fusion. It is easy to show that with equal weights, Eq. (17) degenerates to the diffusion step of NF in Eq. (2). One subtle identity is needed for the equivalence, *i.e.*, $\alpha = 1/\Lambda$. According to the definition of Λ in Eq. (15), $1-\alpha = \gamma/\Lambda$.

Tensor Product Fusion. The similarity propagation in Eq. (16) can be transformed into

$$\begin{aligned}&(\sum_{\nu=1}^M \beta_\nu \mathbf{S}^\nu)\mathbf{A}^{(t)}(\sum_{\mu=1}^M \beta_\mu \mathbf{S}^\mu) = \\ &\underbrace{\sum_{\mu=1}^M \beta_\mu^2 \mathbf{S}^\mu \mathbf{A}^{(t)} \mathbf{S}^\mu}_{\text{RED}} + \underbrace{\sum_{\mu\neq\nu}^M \beta_\mu\beta_\nu \mathbf{S}^\nu \mathbf{A}^{(t)} \mathbf{S}^\mu}_{\text{TPF}}.\end{aligned} \tag{28}$$

By substituting the simplex in Eq. (26) into Eq. (28) and selecting the μ-th and the ν-th affinity graph, we can obtain the fusion with diffusion step of TPF in Eq. (4) by defining $\alpha = 1/\Lambda$.

Regularized Ensemble Diffusion. By substituting the simplex in Eq. (27) into Eq. (28), Eq. (16) becomes

$$\mathbf{A}^{(t+1)} = \frac{1}{\Lambda}\sum_{\mu=1}^M \beta_\mu^2 \mathbf{S}^\mu \mathbf{A}^{(t)} \mathbf{S}^\mu + \frac{\gamma}{\Lambda}\mathbf{I}, \tag{29}$$

which is equivalent to the diffusion step of Eq. (6) if considering β_μ^2 $(1 \leq \mu \leq M)$ as the target weight to be learned.

Finally, it should be mentioned that the fusion step of weight learning varies with different methods.

4.3. Summary of Main Contributions

As summarized in Sec. 2.4, existing fusion methods have different pros and cons. In comparison, UED inherits the advantages and discards the disadvantages with a delicate design of objective function and derivation.

First, the diffusion step of UED is almost as fast as naive fusion. As Eq. (9) shows, it can also merge multiple input similarities in one trial, and does not need to exhaustively apply diffusion step to each input as tensor product fusion and regularized ensemble diffusion. Second, we can draw from Eq. (28) that UED can also consider the interplay of two distinct affinity graphs as tensor product fusion, so that the complementarity between metrics can be better exploited. More importantly, UED is not limited to only fusing two inputs as tensor product fusion. Instead, it can also tackle more than two input similarities as naive fusion

Baselines	ModelNet40		ModelNet10	
	AUC	mAP	AUC	mAP
B1	77.19	76.52	88.97	87.98
B2	80.12	79.41	89.02	88.17
B3	80.39	79.53	91.24	89.97
B4	45.10	44.52	62.37	61.47

Table 1. The performance (%) of four baselines on the ModelNet40 and ModelNet10 dataset.

and regularized ensemble diffusion. Third, due to the dynamic weight learning paradigm, UED is robust to noisy input similarities. Meanwhile, to tackle the non-convex optimization, we also introduce replicator equation as an effective optimizer for weight learning.

At last, we emphasize that UED is not merely an algorithm about metric fusion in re-ranking. More importantly, it can summarize existing methods in a unified framework with a theoretically-sound explanation.

5. Experiments

In this section, we evaluate the proposed framework on various retrieval tasks, including 3D shape retrieval, image retrieval, and person re-identification.

5.1. 3D Shape Retrieval

3D shape retrieval has been an important topic in 3D vision especially in recent years. The experimental comparison is done on the ModelNet dataset [60], which is a representative large-scale 3D shape repository. The current version of ModelNet consists of $151,128$ 3D CAD models, divided into 662 object categories. Following [55, 6], we use two subsets to evaluate the retrieval performance, *i.e.*, ModelNet40, containing $12,311$ shapes in 40 object categories, and ModelNet10, containing $4,899$ shapes in 10 object categories. We use the same training-testing split as in [6, 24, 55, 53, 18] and employ Area Under precision-recall Curve (AUC) and mean Average Precision (mAP) as the evaluation metrics.

Baselines. In order to ensure a fair comparison, we adopt exactly the same four baseline similarity measures as in [8], including GIFT [6, 7], ResNet [17], Volumetric CNN [43], and PANORAMA [38]. For the notation simplification, we denote them as **B1**, **B2**, **B3**, and **B4**, respectively. The baseline performance is presented in Table 1.

Comparison with Fusion Methods. In Tables 2 and 3, we compare the results of those fusion with diffusion methods summarized in the proposed framework on the ModelNet40 and ModelNet10 datasets, respectively. As TPF can only fuse two similarities each time, its results are given in a range. The evaluation is done by fusing the 3-combination of the similarity sets or all the four similarities.

As can be drawn from Table 2, the proposed UED obtains the best performance in most similarity combinations on the ModelNet40 dataset. For example, when fusing **B2**, **B3**, and **B4**, UED reports AUC 88.05 and mAP 87.30. In terms of AUC, the reported performance is better than RED by 1.57, the best trial of TPF by 2.05, and NF by 3.41, respectively. In terms of mAP, UED outperforms RED by 1.59, the best trial of TPF by 2.18, and NF by 3.37, respectively. It firmly testifies that UED can inherit the merits of existing fusion with diffusion methods to learn a more robust similarity.

An abnormal case arises when fusing **B1**, **B2**, and **B3**, where UED only achieves AUC 87.27 and mAP 86.55, a comparable performance with the best competitor NF. As analyzed above, NF is vulnerable to noisy similarities. Nevertheless, Table 1 presents that **B1**, **B2**, and **B3** have very similar performances, while the performance of **B4** is much inferior, indicating that much more noisy edges are involved in the affinity graph parameterized by **B4**. Therefore, when **B4** is involved, NF fails to work well due to the lack of a weight learning mechanism to mitigate the negative influence of noise. By contrast, combining **B1**, **B2**, and **B3** using equal weights is justified, and NF is a cheap solution in this situation.

In Table 4, we present the weights learned by RED and UED. In RED [8], the weight of **B4** is set to 0 in order to totally eliminate its negative contribution to the similarity learning. However, in UED, the weight of **B4** is 0.014, a small but non-zero value. Such a difference originates from the fact that RED fuses multiple similarities by considering each input similarity individually, while UED is able to consider the interplay of two distinct similarities as shown in Eq. (28). Even though **B4** brings in more noisy edges, it can still provide complementary information if integrated with other heterogeneous similarities.

Comparison with State-of-the-arts. Table 5 gives a thorough comparison with state-of-the-art methods on the ModelNet dataset. The results are quoted from the leaderboard of ModelNet, available at `http://modelnet.cs.princeton.edu/`.

As can be observed from the table, UED achieves the best AUC and the second best mAP on both datasets. As a view-based algorithm, SeqViews2SeqLabels [16] proposes an encoder-decoder RNN structure with attention to aggregate the sequential views and reports the best mAP 89.09 on the ModelNet40 dataset. Meanwhile, PANORAMA-ENN [49] is an extension of PANORAMA-NN [50] which uses the panoramic views for model training. It further exploits a new 3-channel schema representation and an ensemble of multiple models, then achieves the best mAP 93.28 on the ModelNet10 dataset. Nevertheless, as an algorithm about re-ranking and metric fusion, it can be anticipated that UED can lead to a better performance if fusing

Baselines	AUC				mAP			
	NF	TPF	RED	Ours	NF	TPF	RED	Ours
B1+B2+B3	**87.53**	83.99∼86.00	87.04	87.27	**86.77**	83.15∼85.12	86.30	86.55
B1+B2+B4	80.02	68.56∼84.01	83.60	**84.70**	79.32	67.16∼83.23	82.82	**83.92**
B1+B3+B4	83.54	68.56∼84.79	85.06	**86.29**	82.83	67.16∼83.86	84.24	**85.38**
B2+B3+B4	84.64	70.69∼86.00	86.48	**88.05**	83.93	69.15∼85.12	85.71	**87.30**
B1+B2+B3+B4	85.26	68.56∼86.00	87.03	**87.22**	84.55	67.16∼85.12	86.30	**86.50**

Table 2. The performance comparison (%) of fusion methods on the ModelNet40 dataset.

Baselines	AUC				mAP			
	NF	TPF	RED	Ours	NF	TPF	RED	Ours
B1+B2+B3	92.80	91.63∼92.60	93.20	**93.37**	91.65	90.56∼91.48	92.15	**92.26**
B1+B2+B4	91.45	84.34∼92.38	92.65	**92.85**	90.25	82.85∼91.41	91.50	**91.74**
B1+B3+B4	91.35	83.97∼92.60	93.23	**93.27**	90.03	82.56∼91.48	**92.17**	92.08
B2+B3+B4	90.67	83.97∼92.14	92.35	**92.49**	89.71	82.56∼91.11	91.23	**91.41**
B1+B2+B3+B4	91.72	83.97∼92.60	93.20	**93.36**	90.49	82.56∼91.48	92.15	**92.25**

Table 3. The performance comparison (%) of fusion methods on the ModelNet10 dataset.

Methods	**B1**	**B2**	**B3**	**B4**
RED	0.356	0.348	0.296	0.000
UED	0.335	0.336	0.312	0.014

Table 4. The learned weights on the ModelNet40 dataset.

Methods	ModelNet40		ModelNet10	
	AUC	mAP	AUC	mAP
SPH [25]	34.47	33.26	45.97	44.05
LFD [10]	42.04	40.91	51.70	49.82
PANORAMA [38]	45.00	46.13	60.72	60.32
ShapeNets [60]	49.94	49.23	69.28	68.26
Geometry Image [54]	-	51.30	-	74.90
DeepPano [53]	77.63	76.81	85.45	84.18
MVCNN [55]	-	79.50	-	-
GIFT [6]	83.10	81.94	92.35	91.12
PANORAMA-NN [50]	-	83.45	-	87.39
GVCNN [13]	-	85.70	-	-
RED [8]	**87.03**	86.30	**93.20**	92.15
PANORAMA-ENN [49]	-	86.34	-	**93.28**
SeqViews2SeqLabels [16]	-	**89.09**	-	91.43
UED (**ours**)	**88.05**	**87.30**	**93.37**	**92.26**

Table 5. The performance comparison (%) with state-of-the-arts on the ModelNet40 and ModelNet10 dataset. The best and second best results are marked in red and blue, respectively.

SeqViews2SeqLabels [16] and PANORAMA-ENN [49] as the input similarities.

5.2. Image Retrieval

We then evaluate the retrieval performance on the Holidays [23] dataset. Holidays dataset is a widely-used benchmark dataset for image retrieval, which is comprised of $1,491$ images and 500 queries. The evaluation metric is mean Average Precision (mAP). Four baseline similarities are used, including NetVLAD [1]: mAP 88.29, SPoC [2]: mAP 86.07, ResNet [17]: mAP 81.83, and HSV color histogram [69]: mAP 61.83. We denote them by **B1**, **B2**, **B3**, and **B4**, respectively in Table 6.

Baselines	NF	TPF	RED	Ours
B1+B2+B3	92.43	90.03∼92.46	**93.32**	93.31
B1+B2+B4	90.65	87.36∼92.46	93.09	**93.13**
B1+B3+B4	89.85	85.12∼91.87	92.55	**93.22**
B2+B3+B4	88.91	85.12∼90.12	90.34	**90.37**
B1+B2+B3+B4	90.69	85.12∼92.46	93.32	**93.56**

Table 6. The performance comparison of different fusion methods on the Holidays dataset.

In line with previous experiments, UED beats NF, TPF, and RED with all but one similarity combinations as presented in Table 6. Meanwhile, by simply fusing four baseline similarities in the re-ranking stage, UED achieves mAP 93.56 on the Holidays dataset. This achievement is already better than the state-of-the-art methods, including Pairwise Geometric Matching [28]: 89.2, Gordo *et al*. [14]: 89.1, Iscen *et al*. [21]: 87.5, Radenović *et al*. [45]: 82.5, and only slightly inferior to Gordo *et al*. [15]: 94.8. However, it can be envisioned that the performance of UED can be better if more discriminative features [15, 36, 39] and an ensemble of models [20, 22, 44] are used.

Here, we do not report the experimental results on the UKbench dataset [35], because the performance on it has already gotten saturated. With the upper bound of the per-

formance being N-S score 4, some previous works have reported nearly perfect scores. For example, Gordo *et al*. [15] report 3.91 by enhancing R-MAC descriptor [58]. Therefore, we include the comparison on the Ukbench dataset in the supplementary material.

5.3. Person Re-identification

In recent years, person re-identification (re-ID) has attracted much attention in the vision community, driven by the demand of video surveillance. Particularly, re-ranking-based approaches [71, 48, 32, 65, 64, 29] become a popular tool to automatically refine the search results.

In this section, we evaluate the proposed method on the Market-1501 dataset [70]. Market-1501 is a widely-used large scale benchmark for person re-identification. It consists of 1501 identities. 750 identities (12,936 images) are used for training, 751 identities (19,732 images) are used for testing, and 3,368 images act as queries. We utilize three baseline similarities. First, we finetune a ResNet-50 model [17] with softmax loss and triplet loss [19]. Then, we extract the L_2 normalized activations of networks before the loss layer as image features and compute the Euclidean distance to measure the similarities between images. We denote the two baselines as **B1** and **B2** respectively. Moreover, Mancs [59], a recent work using attention mechanism, acts as the 3rd baseline similarity **B3**. The performance is measured via rank-1 accuracy and mean Average Precision (mAP) in single-query setting. The baseline performances of **B1**, **B2**, and **B3** are 91.66, 89.22, and 93.17 in rank-1 accuracy, 78.90, 75.33, and 82.51 in mAP, respectively.

Since massive works have reported performance on the Market-1501 dataset, it is simply intractable to compare all of them. Hence, we only include the state-of-the-art methods published in the year 2018 and those about re-ranking or metric fusion in Table 7. Among them, K-reciprocal [71], SSM [3], PSE+ECN [48], and RED [8] are also re-ranking-based approaches as ours. We also reproduce the results of K-reciprocal and RED with publicly available codes using the same baselines to ensure a fair comparison. Since K-reciprocal can only handle one feature, we concatenate multiple features as its input. As can be drawn from the table, the results (either original or reproduced ones) of the re-ranking algorithms are all inferior to that of UED. In Fig. 1, we give a qualitative evaluation by exhibiting several probe images and their 1-nearest neighbors with a disjoint camera ID. The matching pairs are correctly retrieved by UED, while RED* and K-reciprocal* fail to identify these persons.

UED also outperforms some latest representatives by a large margin, including AWTL [47], HA-CNN [27], and Mancs [59]. Moreover, UED achieves mAP 92.75, which is the first work reporting mAP larger than 90 to our best knowledge. In this sense, it will be a feasible way to improve the recognition rate of re-ID systems by using model ensemble and re-ranking in the future work.

Methods	Rank-1 Accuracy	mAP
AWTL [47]	89.46	75.67
HA-CNN [27]	91.20	75.70
Mancs [59]	93.17	82.51
K-reciprocal [71]	77.11	63.63
SSM [3]	82.21	68.80
PSE+ECN [48]	90.30	84.00
RED* [8]	94.74	91.00
K-reciprocal* [71]	94.69	91.87
UED (**ours**)	**95.90**	**92.75**

Table 7. The performance comparison (%) on the Market-1501 dataset. The results marked with $*$ are reproduced with publicly available codes using the same baselines.

Figure 1. Example matching pairs of probe and gallery images correctly retrieved by UED on the Market-1501 dataset.

6. Conclusion

In this paper, we have concentrated on re-ranking with the capacity of metric (or similarity) fusion for object retrieval and person re-identification. The proposed Unified Ensemble Diffusion (UED) is not only an effective algorithm which achieves the state-of-the-art retrieval performance on benchmark datasets, but also a unified and theoretical framework, within which existing fusion methods are summarized as its special cases. By deeply analyzing the principles of existing fusion methods, UED has undergone a careful design of objective function and derivation, which enables it to have a fast diffusion step, consider the interplay of all input pairs, handle multiple inputs, and be robust to noise.

Most current re-ranking methods are not end-to-end trainable, only serving as a post-processing procedure to refine the retrieval results. Recently, several works [51, 52] have suggested to construct the affinity graph in a mini-batch in a deep model and achieved a promising performance improvement. However, it is difficult to well sample the manifold structure given a small set of data points. Therefore, how to include the contextual information in a mini-batch is still an open-problem. We leave this as our future work.

Acknowledgements This work was supported by the EPSRC grant Seebibyte EP/M013774/1, EPSRC/MURI grant EP/N019474/1 and NSF grant IIS-1814745. We would also like to acknowledge the Royal Academy of Engineering and FiveAI.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 7

[2] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *ICCV*, pages 1269–1277, 2015. 7

[3] S. Bai, X. Bai, and Q. Tian. Scalable person re-identification on supervised smoothed manifold. In *CVPR*, 2017. 8

[4] S. Bai, X. Bai, Q. Tian, and L. J. Latecki. Regularized diffusion process for visual retrieval. In *AAAI*, pages 3967–3973, 2017. 1, 2

[5] S. Bai, X. Bai, Q. Tian, and L. J. Latecki. Regularized diffusion process on bidirectional context for object retrieval. *TPAMI*, 2019. 1, 2

[6] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki. Gift: A real-time and scalable 3d shape search engine. In *CVPR*, 2016. 6, 7

[7] S. Bai, X. Bai, Z. Zhou, Z. Zhang, Q. Tian, and L. J. Latecki. Gift: Towards scalable 3d shape retrieval. *TMM*, 19(6):1257–1271, 2017. 6

[8] S. Bai, Z. Zhou, J. Wang, X. Bai, L. J. Latecki, and Q. Tian. Ensemble diffusion for retrieval. In *ICCV*, pages 774–783, 2017. 1, 3, 6, 7, 8, 11

[9] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96, 2005. 2

[10] D. Y. Chen, X. P. Tian, Y. T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Comput. Graph. Forum*, 22(3):223–232, 2003. 7

[11] M. Donoser and H. Bischof. Diffusion processes for retrieval revisited. In *CVPR*, pages 1320–1327, 2013. 1, 2

[12] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong. 3d deep shape descriptor. In *CVPR*, pages 2319–2328, 2015. 1

[13] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *CVPR*, pages 264–272, 2018. 7

[14] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, pages 241–257, 2016. 7

[15] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 124(2):237–254, 2017. 7, 8

[16] Z. Han, M. Shang, Z. Liu, C.-M. Vong, Y.-S. Liu, M. Zwicker, J. Han, and C. P. Chen. Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention. *TIP*, 28(2):658–672, 2019. 6, 7

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7, 8

[18] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai. Triplet-center loss for multi-view 3d object retrieval. In *CVPR*, 2018. 6

[19] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 8

[20] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chum. Fast spectral ranking for similarity search. In *CVPR*, 2018. 7

[21] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Mining on manifolds: Metric learning without labels. In *CVPR*, 2018. 7

[22] A. Iscen, G. Tolias, Y. S. Avrithis, T. Furon, and O. Chum. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *CVPR*, 2017. 1, 7

[23] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, pages 304–317, 2008. 2, 7

[24] E. Johns, S. Leutenegger, and A. J. Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *CVPR*, 2016. 6

[25] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP*, pages 156–164, 2003. 7

[26] Q. Ke and Y. Li. Is rotation a nuisance in shape recognition? In *CVPR*, pages 4146–4153, 2014. 1

[27] W. Li, X. Zhu, and S. Gong. Harmonious attention network for person re-identification. In *CVPR*, 2018. 8

[28] X. Li, M. Larson, and A. Hanjalic. Pairwise geometric matching for large-scale object retrieval. In *CVPR*, pages 5153–5161, 2015. 7

[29] C. Liu, C. Change Loy, S. Gong, and G. Wang. Pop: Person re-identification post-rank optimisation. In *ICCV*, pages 441–448, 2013. 8

[30] M. Liu, B. C. Vemuri, S. ichi Amari, and F. Nielsen. Shape retrieval using hierarchical total bregman soft clustering. *TPAMI*, 34(12):2407–2419, 2012. 1

[31] V. Losert and E. Akin. Dynamics of games and genes: Discrete versus continuous time. *Journal of Mathematical Biology*, 17(2):241–251, 1983. 4

[32] C. C. Loy, C. Liu, and S. Gong. Person re-identification by manifold ranking. In *ICIP*, pages 3567–3571, 2013. 8

[33] L. Luo, C. Shen, C. Zhang, and A. van den Hengel. Shape similarity analysis by self-tuning locally constrained mixed-diffusion. *TMM*, 15(5):1174–1183, 2013. 1

[34] T. Mei, Y. Rui, S. Li, and Q. Tian. Multimedia search reranking: A literature survey. *ACM Comput. Surv.*, 46(3):38:1–38:38, 2014. 1

[35] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006. 2, 7, 11

[36] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large scale image retrieval with attentive deep local features. In *ICCV*, pages 3456–3465, 2017. 7

[37] S. Paisitkriangkrai, C. Shen, and A. Van Den Hengel. Learning to rank in person re-identification with metric ensembles. In *CVPR*, pages 1846–1855, 2015. 1, 2

[38] P. Papadakis, I. Pratikakis, T. Theoharis, and S. J. Perantonis. Panorama: A 3d shape descriptor based on panoramic views

for unsupervised 3d object retrieval. *IJCV*, 89(2-3):177–192, 2010. 1, 6, 7

[39] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. In *ICCV*, pages 91–99, 2015. 7

[40] D. C. G. Pedronette and R. D. S. Torres. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, 46(8):2350–2360, 2013. 1

[41] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11(8):1933–1955, 1999. 2, 4

[42] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. In *NIPS*, pages 550–556, 1999. 2, 4, 11

[43] C. R. Qi, H. Su, M. Niessner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *CVPR*, 2016. 6

[44] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 7

[45] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016. 7

[46] B. Ramesh, C. Xiang, and T. H. Lee. Shape classification using invariant features and contextual information in the bag-of-words model. *Pattern Recognition*, 48(3):894–906, 2015. 1

[47] E. Ristani and C. Tomasi. Features for multi-target multi-camera tracking and re-identification. In *CVPR*, 2018. 8

[48] M. Saquib Sarfraz, A. Schumann, A. Eberle, and R. Stiefelhagen. A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. In *CVPR*, 2018. 8

[49] K. Sfikas, I. Pratikakis, and T. Theoharis. Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval. *Computers & Graphics*, 71:208–218, 2018. 1, 6, 7

[50] K. Sfikas, T. Theoharis, and I. Pratikakis. Exploiting the panorama representation for convolutional neural network classification and retrieval. In *3DOR*, 2017. 6, 7

[51] Y. Shen, H. Li, T. Xiao, S. Yi, D. Chen, and X. Wang. Deep group-shuffling random walk for person re-identification. In *CVPR*, pages 2265–2274, 2018. 8

[52] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang. Person re-identification with deep similarity-guided graph neural network. In *ECCV*, pages 508–526, 2018. 8

[53] B. Shi, S. Bai, Z. Zhou, and X. Bai. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343, 2015. 6, 7

[54] A. Sinha, J. Bai, and K. Ramani. Deep learning 3d shape surfaces using geometry images. In *ECCV*, pages 223–240, 2016. 7

[55] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, pages 945–953, 2015. 6, 7

[56] H. Tabia, M. Daoudi, J.-P. Vandeborre, and O. Colot. A new 3d-matching method of nonrigid and partially similar models using curve analysis. *TPAMI*, 33(4):852–858, 2011. 1

[57] H. Tabia, H. Laga, D. Picard, and P.-H. Gosselin. Covariance descriptors for 3d shape matching and retrieval. In *CVPR*, pages 4185–4192, 2014. 1

[58] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *ICLR*, 2015. 8

[59] C. Wang, Q. Zhang, C. Huang, W. Liu, and X. Wang. Mancs: A multi-task attentional network with curriculum sampling for person re-identification. In *ECCV*, pages 365–381, 2018. 8

[60] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shape modeling. In *CVPR*, 2015. 2, 6, 7

[61] J. Xie, Y. Fang, F. Zhu, and E. Wong. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In *CVPR*, pages 1275–1283, 2015. 1

[62] F. Yang, B. Matei, and L. S. Davis. Re-ranking by multi-feature fusion with diffusion for image retrieval. In *WACV*, pages 572–579, 2015. 1

[63] X. Yang, L. Prasad, and L. J. Latecki. Affinity learning with diffusion on tensor product graph. *TPAMI*, 35(1):28–38, 2013. 2

[64] M. Ye, C. Liang, Y. Yu, Z. Wang, Q. Leng, C. Xiao, J. Chen, and R. Hu. Person reidentification via ranking aggregation of similarity pulling and dissimilarity pushing. *TMM*, 18(12):2553–2566, 2016. 8

[65] M. Ye, A. J. Ma, L. Zheng, J. Li, and P. C. Yuen. Dynamic label graph matching for unsupervised video re-identification. In *ICCV*, 2017. 8

[66] T. Yu, Y. Wu, S. D. Bhattacharjee, and J. Yuan. Efficient object instance search using fuzzy objects matching. In *AAAI*, pages 4320–4326, 2017. 1

[67] T. Yu, J. Yuan, C. Fang, and H. Jin. Product quantization network for fast image retrieval. In *ECCV*, pages 186–201, 2018. 1

[68] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *ECCV*, pages 660–673, 2012. 1

[69] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific rank fusion for image retrieval. *TPAMI*, 37(4):803–815, 2015. 1, 7

[70] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, pages 1116–1124, 2015. 2, 8

[71] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, pages 3652–3661, 2017. 8

[72] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS*, pages 169–176, 2004. 1, 2

[73] Y. Zhou, X. Bai, W. Liu, and L. J. Latecki. Similarity fusion for visual tracking. *IJCV*, pages 1–27, 2016. 1, 2

Appendices

The below appendices contain the supplementary material for "Re-ranking via Metric Fusion for Object Retrieval

and Person Re-identification". The proofs of two key statements made in the main manuscript are given in Sec. A. The additional performance evaluation and comparisons are given in Sec. B.

A. Proofs

Proposition 1. *Eq. (19) converges to the closed-form solution in Eq. (14).*

Proof. By executing the iteration for t times, $\vec{A}^{(t+1)}$ can be expanded as

$$\tilde{A}^{(t+1)} = (\frac{\mathbf{S}}{\Lambda})^t \tilde{A}^{(1)} + \frac{\gamma}{\Lambda}\sum_{i=0}^{t-1}(\frac{\mathbf{S}}{\Lambda})^i \tilde{I}, \tag{30}$$

where

$$\mathbf{S} = \sum_{\mu,\nu=1}^{M} \beta_\mu \beta_\nu (\mathbf{S}^\mu \otimes \mathbf{S}^\nu). \tag{31}$$

It is known that the spectral radius of both $\mathbf{S}^\mu$ and $\mathbf{S}^\nu$ are no larger than 1. According to the spectral property of Kronecker product, all the eigenvalues of $\mathbf{S}^\mu \otimes \mathbf{S}^\nu$ are also in [-1,1]. Hence, the spectral radius of $\mathbf{S}/\Lambda$ is bounded by

$$\frac{1}{\Lambda}\sum_{\mu,\nu=1}^{M} \beta_\mu \beta_\nu = \frac{1}{\Lambda} = \frac{1}{\gamma+1} < 1. \tag{32}$$

Recall that $\gamma > 0$. Then, we have

$$\begin{aligned} &\lim_{t\to\infty}(\frac{\mathbf{S}}{\Lambda})^t = 0, \\ &\lim_{t\to\infty}\sum_{i=0}^{t-1}(\frac{\mathbf{S}}{\Lambda})^i = (\mathbf{I} - \frac{\mathbf{S}}{\Lambda})^{-1}. \end{aligned} \tag{33}$$

As a result, we derive that

$$\begin{aligned} \lim_{t\to\infty} \tilde{A}^{(t+1)} &= \frac{\gamma}{\Lambda}(\mathbf{I} - \frac{\mathbf{S}}{\Lambda})^{-1}\tilde{I} \\ &= \frac{\gamma}{\Lambda}(\mathbf{I} - \frac{1}{\Lambda}\sum_{\mu,\nu=1}^{M} \beta_\mu \beta_\nu \mathbf{S}^\mu \otimes \mathbf{S}^\nu)^{-1}\vec{\mathbf{I}}, \end{aligned} \tag{34}$$

which is equivalent to Eq. (14). The proof is complete. □

Proposition 2. *The minimization in Eq. (20) is equivalent to the maximization in Eq. (21).*

Recall that the objective function of Eq. (20) is

$$\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\mathbf{H}\boldsymbol{\beta} + \eta\|\boldsymbol{\beta}\|_2^2, \;\; s.t.\; \boldsymbol{\beta} \in \Delta, \tag{35}$$

and the objective function of Eq. (21) is

$$\max_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\bar{\mathbf{H}}\boldsymbol{\beta}, \;\; s.t.\; \boldsymbol{\beta} \in \Delta, \tag{36}$$

where $\bar{\mathbf{H}} = -\mathbf{H}/2 - \mathbf{H}^{\mathrm{T}}/2 - \eta\mathbf{I} + \mathbf{C}$ and $\mathbf{C} \in \mathbb{R}^{M\times M}$ is a matrix with all its entries equal to the maximum element of $(\mathbf{H}/2 + \mathbf{H}^{\mathrm{T}}/2 + \eta\mathbf{I})$.

Proof. To prove the equivalence, first we have the following preliminary fact

$$\boldsymbol{\beta}^{\mathrm{T}}\frac{\mathbf{H} - \mathbf{H}^{\mathrm{T}}}{2}\boldsymbol{\beta} \equiv 0. \tag{37}$$

It holds, since $(\mathbf{H}/2 - \mathbf{H}^{\mathrm{T}}/2)$ is an antisymmetric matrix.

Then, we have

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\mathbf{H}\boldsymbol{\beta} + \eta\|\boldsymbol{\beta}\|_2^2 \\ \Leftrightarrow &\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\frac{\mathbf{H} + \mathbf{H}^{\mathrm{T}}}{2}\boldsymbol{\beta} + \boldsymbol{\beta}^{\mathrm{T}}\frac{\mathbf{H} - \mathbf{H}^{\mathrm{T}}}{2}\boldsymbol{\beta} + \eta\|\boldsymbol{\beta}\|_2^2 \\ \Leftrightarrow &\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\frac{\mathbf{H} + \mathbf{H}^{\mathrm{T}}}{2}\boldsymbol{\beta} + \eta\|\boldsymbol{\beta}\|_2^2 \\ \Leftrightarrow &\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}(\mathbf{H}/2 + \mathbf{H}^{\mathrm{T}}/2 + \eta\mathbf{I})\boldsymbol{\beta} \\ \Leftrightarrow &\max_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}(-\mathbf{H}/2 - \mathbf{H}^{\mathrm{T}}/2 - \eta\mathbf{I})\boldsymbol{\beta}. \end{aligned} \tag{38}$$

As replicator equation [42] requires non-negative input, we define $\mathbf{C} \in \mathbb{R}^{M\times M}$ is a matrix with all its entries equal to the maximum element of $(\mathbf{H}/2 + \mathbf{H}^{\mathrm{T}}/2 + \eta\mathbf{I})$. It is easy to see that $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{C}\boldsymbol{\beta}$ is a constant. Then, Eq. (38) is tranformed into

$$\begin{aligned} &\max_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}(-\mathbf{H}/2 - \mathbf{H}^{\mathrm{T}}/2 - \eta\mathbf{I} + C)\boldsymbol{\beta} \\ \Leftrightarrow &\max_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\bar{\mathbf{H}}\boldsymbol{\beta}, \end{aligned} \tag{39}$$

which is equivalent to Eq. (21). The proof is complete. □

B. Experiment on Ukbench

Ukbench dataset [35] is a classical and representative benchmark for image retrieval, which is composed of $10,200$ images. The whole dataset has $2,550$ categories with 4 images per category. Each image is used in turn as a query. The performance is measured by the average recall of the top-4 ranked images, referred as N-S score (maximum is 4). In recent years, the performance on the Ukbench dataset has gradually gotten saturated. Therefore, we do not include the performance comparison in the main manuscript. As can be drawn from Table 8, compared with RED [8], the proposed UED achieves better performance in three settings and the same performance in two settings.

Baselines	NF	TPF	RED	Ours
B1+B2+B3	3.900	3.854∼3.884	**3.919**	**3.919**
B1+B2+B4	3.822	3.626∼3.876	3.920	**3.922**
B1+B3+B4	3.865	3.626∼3.884	3.927	**3.930**
B2+B3+B4	3.893	3.629∼3.861	3.923	**3.926**
B1+B2+B3+B4	3.907	3.626∼3.884	**3.938**	**3.938**

Table 8. The performance comparison of different fusion methods on the Ukbench dataset.