# Regularized Diffusion Process on Bidirectional Context for Object Retrieval

Song Bai, *Student Member, IEEE,* Xiang Bai, *Senior Member, IEEE,*
Qi Tian, *Fellow, IEEE,* and Longin Jan Latecki, *Senior Member, IEEE.*

**Abstract**—Diffusion process has advanced object retrieval greatly as it can capture the underlying manifold structure. Recent studies have experimentally demonstrated that tensor product diffusion can better reveal the intrinsic relationship between objects than other variants. However, the principle remains unclear, *i.e.*, what kind of manifold structure is captured. In this paper, we propose a new affinity learning algorithm called Regularized Diffusion Process (RDP). By deeply exploring the properties of RDP, our first yet basic contribution is providing a manifold-based explanation for tensor product diffusion. A novel criterion measuring the smoothness of the manifold is defined, which simultaneously regularizes four vertices in the affinity graph. Inspired by this observation, we further contribute two variants towards two specific goals. While ARDP can learn similarities across heterogeneous domains, HRDP performs affinity learning on tensor product hypergraph, considering the relationships between objects are generally more complex than pairwise. Consequently, RDP, ARDP and HRDP constitute a generic tool for object retrieval in most commonly-used settings, no matter the input relationships between objects are derived from the same domain or not, and in pairwise formulation or not. Comprehensive experiments on 10 retrieval benchmarks, especially on large scale data, validate the effectiveness and generalization of our work.

**Index Terms**—Image Retrieval, 3D Shape Retrieval, Cross-modal Retrieval, Affinity Learning, Re-ranking, Diffusion Process.

## 1 Introduction

Given a query object, the goal of retrieval task is to return similar objects in the database according to a pre-defined similarity measure. Conventionally, it is accomplished by computing the pairwise dissimilarity between features in the Euclidean space. Then, similar objects are expected to be distributed with larger similarities to the query. Thus, they can be ranked in higher positions of the ranking list. However, it has been demonstrated [1] that the pairwise formulation is insufficient to reveal the intrinsic relationship between objects. Instead, similarities can be estimated more accurately along the geodesic path of the underlying data manifold, *i.e.*, in the context of other objects.

To illustrate the concept, we present a toy example in Fig. 1. The data distribution is a two-spiral pattern with 200 data points, with each spiral having 100 points and one query point in cross shape. An ideal retrieval result is that points in one spiral have larger similarities with the query in this spiral than the query in the other spiral. The Euclidean distance (see Fig. 1a) is inadequate, while the proposed method (see Fig. 1b) is able to reveal the data structure.

To capture the geometry structure of the manifold, many algorithms have been developed in the literature. Those algorithms share a very diverse nomenclature, including but not limited to context sensitive similarity [2], [3], affinity learning [4], [5], re-ranking [6], [7], [8], [9], ranking list comparison [10], [11], [12]. Nevertheless, most of them model the relationship between objects on a graph-based manifold, where the vertices in the graph represent objects and the edge connecting two adjacent vertices is weighted by their similarity. Then, similarity values are diffused on the graph in an iterative manner (*e.g.*, random walk [13]). This procedure is usually called diffusion process [14], [15] in the retrieval domain.

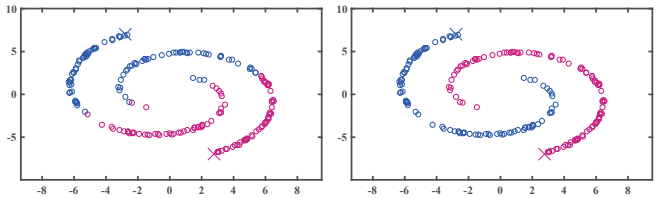


Fig. 1. The retrieval results returned by the Euclidean distance (a) and the proposed algorithm (b). The two crosses denote the query points. All other points are colored according to the larger similarity to one of the two query points.

Most existing algorithms focus on iteration-based models, with differences in similarity initialization, transition matrix initialization and iteration scheme. A recent survey paper [1] summarizes most common variants of diffusion process in a unified framework, and provides a strong experimental support for those iteration-based models in terms of retrieval performance. According to its taxonomy, diffusion process on tensor product graph [16], built by computing the tensor product of the original affinity graph with itself, exhibits its superiority over other kinds of diffusion process. Tensor product graph naturally takes into account high order information, which is stated to be helpful for

- *S. Bai and X. Bai are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. E-mail: {songbai, xbai}@hust.edu.cn*
- *Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249-1604 USA. E-mail: qitian@cs.utsa.edu*
- *L.J. Latecki is with the Department of Computer and Information Sciences, Temple University, 1925 N.12th Street, Philadelphia, PA 19122. E-mail: latecki@temple.edu*

*Manuscript received August 8, 2017.*

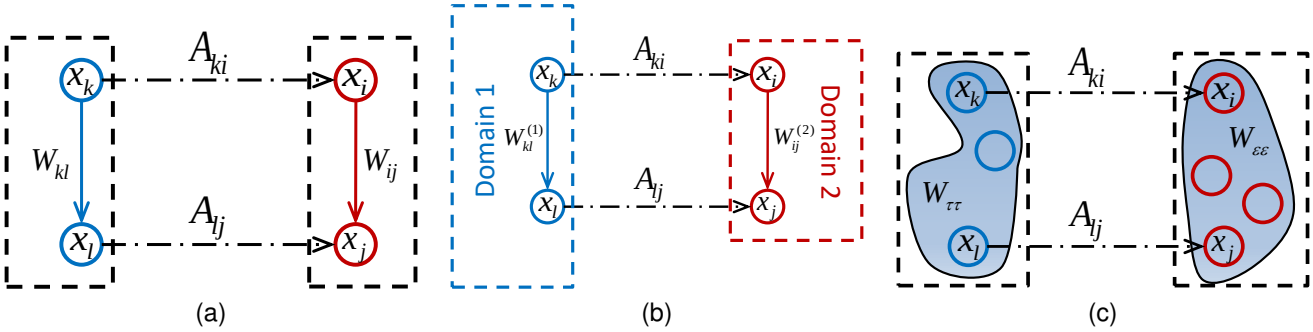


Fig. 2. The illustrations of the smoothness criterion of the proposed RDP (a), ARDP (b) and HRDP (c). $W$ denotes the input similarity, and $A$ denotes the output similarity.

retrieval on manifold. However, no works, including [16] itself and the survey [1], have explained the mechanism behind. Some critical questions are: 1) what kind of manifold structure is captured and why it is better; 2) why high order information is useful; 3) what is the essence of iteration and how many iterations are needed. Unfortunately, though the iteration-based models have been extensively studied (*e.g.*, [1], [16]), those issues are still unexplored. In this sense, the existing investigations about diffusion-based affinity learning remain heuristic and insufficient.

Considering that, the first yet basic contribution of this work is to use regularization-based model to theoretically expose the inherent principle of tensor product graph diffusion, in particular to answer those unexplored questions. To this end, a novel algorithm called Regularized Diffusion Process (**RDP**) is proposed. Though RDP has multiple formulations, its key novelty lies in the regularization framework (Sec. 3.1), which defines a novel smoothness criterion to simultaneously regularize four vertices in the affinity graph (illustrated in Fig. 2a). Instead of heuristically defining the iterative model as [16], we provide strong evidences that regularization can be a better theoretical and practical guidance for tensor product graph diffusion.

By solving the objective function with regularization, one can easily obtain the target similarity. However, it is too computationally demanding to directly use the closed-form solution. To make the computation feasible in practice, we resort to an efficient iteration-based solver (see Sec. 3.2) like existing algorithms [1], [16], [17]. Hence, the essence of the iterative model of RDP is to minimize a kind of relationship among four vertices at each iteration, *i.e.*, to approximate the optimal solution of the regularization framework.

Nevertheless, the proposed RDP, as well as most previous works [1], [14], [15], [16], is only applicable in simple retrieval settings, where the input similarity 1) is within the same data domain, and 2) is in pairwise formulation, limiting its usage in more challenging retrieval situations. Inspired by the regularization framework of RDP, we further contribute two important variants so that the generalization of our work is significantly improved, as

1) **Asymmetric Regularized Diffusion Process (ARDP)** considers affinity learning across two heterogeneous domains (see Fig. 2b). In this case, the context considered in two domains is asymmetric, as the sizes of the two domains are not necessarily equal.

2) **Hypergraph Regularized Diffusion Process (HRDP)** considers that the relationships between objects are more complex than pairwise in many applications. Thus, it essentially performs affinity learning on the tensor product of the hypergraph (see Fig. 2c), where hyperedges are utilized to capture the complex relationships. By doing so, HRDP can leverage the high-order information brought by both the hypergraph itself and the tensor order learning.

With the supplement of ARDP and HRDP, our work is suitable to deal with most commonly-used retrieval settings. Given an input similarity, it not only can learn more faithful similarities than other diffusion-based algorithms, thus yielding better retrieval performances. But more importantly, it can handle more challenging retrieval scenarios, which cannot be handled by [1], [16], [17]. In other words, *we systemically propose a generic and versatile tool for tensor-order affinity learning between objects, with which many previous algorithms can further enhance their retrieval performance*. Meanwhile, as a unified theoretical framework about tensor-order affinity learning is established, we believe that it would be inspiring for other researchers to design algorithms about re-ranking, graph learning and feature fusion, and advance research directions like geometric verification [18], point registration, low-shot learning [19].

To demonstrate the generalization of our work, a series of experiments is conducted. We first verify the effectiveness of RDP in simple retrieval settings but with different data modalities, such as face retrieval on the ORL and the YALE datasets [20], shape retrieval on the MPEG-7 dataset [21], natural image retrieval on the Ukbench [22], the Holidays [23], the Oxford5K [24] and the large scale Oxford105K datasets, and sketch retrieval on the TU Berlin Sketch dataset [25]. Experimental results suggest that RDP can achieve state-of-the-art performances on those datasets as presented from Sec. 5.2 to Sec. 5.4. In Sec. 5.5, we also apply ARDP to cross-modal retrieval on the Wikipedia dataset [26], [27], where retrieval is done between text data and image data. And in Sec. 5.6, the validity of HRDP is testified with view-based 3D model retrieval on the Princeton Shape Benchmark (PSB) [28], where 3D models are connected with hyperedges on a hypergraph.

The rest paper is organized as follows: Sec. 2 reviews the relevant methods. The details of RDP are given in Sec. 3, and its two variants ARDP and HRDP are described in Sec. 4. The experimental comparison and analysis are presented in Sec. 5. The conclusions and the future work are summarized in Sec. 6. All the used theorems, lemmas and definitions are

put in the supplementary material.

## 2 Related Work

Affinity learning between objects is a fundamental topic in computer vision, which has been investigated for decades.

Manifold ranking [17], derived from semi-supervised learning [29], proposes to rank the data with respect to the intrinsic manifold structure. Graph Transduction (GT) [3] takes the query point as the only labeled data, and spreads the labeled information to unlabeled database in a similar way of label propagation. Locally Constrained Diffusion Process (LCDP) [14] further stresses that it is crucial to constrain the diffusion process "locally" since it is susceptible to noise edges in the affinity graph. In [30], an ultra-efficient diffusion process called Regional Diffusion is proposed, which is conducted on descriptors of image regions rather than on global image descriptors considered in this paper. Motivated by the observation that a good ranking is usually asymmetrical, Contextual Dissimilarity Measure (CDM) [2] improves Bag-of-Words (BoW) [31] retrieval system by iteratively estimating the pairwise distance in the spirit of Sinkhorn's scaling algorithm.

Despite those diffusion processes on the original graph, Tensor Product Graph diffusion (TPG) [16] manages leveraging the high-order information from the tensor product of the affinity graph. Its key contribution is that the information propagation on TPG can be computed with the same computational complexity as that on the original graph. The survey [1] defines a general framework for those diffusion processes. By varying $4$ different affinity initializations, $6$ different transition matrices and $3$ different update schemes, it enumerates $72$ variants of diffusion process, and experimentally benchmarks that affinity learning on the tensor product graph is more robust in the scope of retrieval. The previous conference version [32] of this paper theoretically explains why this kind of diffusion process is superior by defining a new smoothness criterion among four vertices.

To leverage the complementarity of multiple cues, large efforts are also devoted to feature fusion in the framework of diffusion process. As a representative work, Graph Fusion [6] integrates multiple features in a query-specific manner, and learns the affinity by utilizing the local PageRank algorithm. Co-transduction [33] combines the concept of co-training and graph transduction for robust shape retrieval. In [34], weight learning and affinity learning are jointly done in a unified framework, which makes it particularly robust to noisy similarities. Locally Constrained Mixed Process [15] partly fuses multiple similarities into one, then propagates on the locally dense data space. In [35], multiple features are combined by a mixture Markov model, and a feature selection method using group sparsity is proposed in [36].

Most aforementioned algorithms are run in an iterative manner. Besides, some methods directly define a new context-sensitive similarity via analyzing the ranking list or the neighborhood structures. For example, Pedronette *et al*. [10], [12] propose a novel similarity measure based on the similarity of the ranking lists. Re-ranking with k-nearest neighbor (kNN) [9], mutual kNN [5] and reciprocal kNN [8], [37] are also explored respectively. The principle of those algorithms is that similar data points tend to have more common neighbors. Though appear different from our work, we demonstrate that they also have inherent connections as presented in Sec. 3.4. Interestingly, diffusion process also relates to dominant sets [38], a well-known graph-theoretic notion which is successfully applied to neighborhood selection [16], image segmentation [39] and geo-localization [40].

## 3 Regularized Diffusion Process

Regularized Diffusion Process (RDP) models the data manifold as an weighted graph $\mathcal{G} = (X, W)$, where the vertices of the graph denote the data points $X = \{x_1, x_2, \ldots, x_N\}$. $W \in \mathbb{R}^{N \times N}$ is the graph adjacency matrix, and $W_{ij}$ represents the pairwise similarity between $x_i$ and $x_j$. Our aim is to learn a new similarity measure $A = \{A_{ij}\}_{1 \le i,j \le N}$, which varies sufficiently smooth along the graph $\mathcal{G}$.

Although modeled on the graph $\mathcal{G}$, the proposed RDP, as expounded below, essentially learns the similarity $A$ on the tensor product graph $\mathbb{G}$ while maintaining the same algorithmic complexity as diffusion on $\mathcal{G}$. In the tensor product graph $\mathbb{G}$, each vertex corresponds to two vertices and each edge depicts the relationship among four vertices in the original graph $\mathcal{G}$. Formally, the tensor product graph $\mathbb{G} = (\mathbb{X}, \mathbb{W})$ is defined as

$$\begin{cases} \mathbb{X} = X \times X, \\ \mathbb{W} = W \otimes W, \end{cases}$$

where $\times$ denotes the Cartesian product and $\otimes$ denotes the Kronecker product.

### 3.1 Regularization Framework

Most pervious works [1] are run in an iterative manner. However, we propose to obtain the new similarity measure $A$ as the closed-form solution of the following optimization problem

$$\begin{aligned} \min_A \frac{1}{2} \sum_{i,j,k,l=1}^{N} W_{ij} W_{kl} \left( \frac{A_{ki}}{\sqrt{D_{ii} D_{kk}}} - \frac{A_{lj}}{\sqrt{D_{jj} D_{ll}}} \right)^2 \\ + \mu \sum_{k,i=1}^{N} (A_{ki} - Y_{ki})^2, \end{aligned} \tag{1}$$

where $\mu > 0$ is a regularization parameter. $Y \in \mathbb{R}^{N \times N}$ denotes the initial affinity values. $D$ is a diagonal matrix with elements $D_{ii} = \sum_{j=1}^{N} W_{ij}$.

As presented in Eq. (1), the objective function of RDP consists of two terms. The first term describes a kind of influence of the input similarity $W$ on the learned similarity $A$. By analogy to Local and Global Consistency (LGC) [29], we will call it *smoothness term*. However, the inherent meanings of two smoothness terms are quite different. As a semi-supervised learning algorithm, the smoothness term of LGC indicates that if $x_k$ is similar to $x_l$ (large $W_{kl}$), their probabilities of belonging to the same category should have a small difference. By contrast, the smoothness term of our method regularizes that if $x_i$ is similar to $x_j$ (large $W_{ij}$) and $x_k$ is also similar to $x_l$ (large $W_{kl}$) in the input similarity space, then the learned similarities $A_{ki}$ and $A_{lj}$ should be similar (see Fig. 2a).

Manifold ranking [17] directly applies LGC to retrieval task by interpreting the probability of belonging to categories as the similarities between objects. Thus, one can find

that the smoothness term in our method actually imposes a relaxed constraint against that in manifold ranking, *i.e.*, the individual object $x_i$ is replaced by a pair of objects $x_i$ and $x_j$ with similarity $W_{ij}$. Consequently, to interrelate four tuples simultaneously, tensor product graph is a natural choice since each of its vertices contains two data points and each of its edges records the relationship between four data points.

In this sense, RDP can be expressed as an extended version of manifold ranking with a relaxed smoothness term. The second term in Eq. (1) is called *fitting term*, which explicitly penalizes the difference from the initial similarity. Previous works [16], [17] take $Y$ as identity matrix $I$, indicating that only the self-affinity of each node is fastened. In the experiments, we verify that this is not an optimal setup.

It seems difficult to derive a closed-form solution of Eq. (1) owing to the difficulty in computing the derivative with respect to $A$. However, our key observation shows that it is possible to transform Eq. (1) so that, it can be easily solved by adapting standard tools from graph theory. To this end, we need two additional operators:

1) $\mathrm{vec}(\cdot)$: vectorize an input matrix by stacking its columns one after the next,
2) $\mathrm{vec}(\cdot)^{-1}$: the inverse operator of $\mathrm{vec}(\cdot)$,

and two identical coordinate transformations $\alpha \equiv N(i-1)+k$ and $\beta \equiv N(j-1)+l$. To simplify the notation, we define $\vec{A} = \mathrm{vec}(A)$ throughout this paper. Then, the smoothness term of Eq. (1) can be transformed into

$$\begin{aligned}
&\frac{1}{2}\sum_{\alpha,\beta=1}^{N^2} \mathbb{W}_{\alpha\beta}\left(\frac{\vec{A}_\alpha}{\sqrt{\mathbb{D}_{\alpha\alpha}}} - \frac{\vec{A}_\beta}{\sqrt{\mathbb{D}_{\beta\beta}}}\right)^2 \\
=&\sum_{\alpha,\beta=1}^{N^2} \mathbb{W}_{\alpha\beta}\frac{\vec{A}_\alpha^2}{\mathbb{D}_{\alpha\alpha}} - \sum_{\alpha,\beta=1}^{N^2} \vec{A}_\alpha \frac{\mathbb{W}_{\alpha\beta}}{\sqrt{\mathbb{D}_{\alpha\alpha}\mathbb{D}_{\beta\beta}}}\vec{A}_\beta \\
=&\sum_{\alpha=1}^{N^2} \vec{A}_\alpha^2 - \vec{A}^{\mathrm{T}}\mathbb{D}^{-1/2}\mathbb{W}\mathbb{D}^{-1/2}\vec{A} \\
=&\vec{A}^{\mathrm{T}}\left(I - \mathbb{D}^{-1/2}\mathbb{W}\mathbb{D}^{-1/2}\right)\vec{A} \\
=&\vec{A}^{\mathrm{T}}(I-\mathbb{S})\vec{A},
\end{aligned} \tag{2}$$

where $I$ is an identity matrix of an appropriate size, $\mathbb{W} = W \otimes W \in \mathbb{R}^{N^2\times N^2}$, $\mathbb{D} = D \otimes D \in \mathbb{R}^{N^2\times N^2}$, $\mathbb{S} = S \otimes S \in \mathbb{R}^{N^2\times N^2}$, and $S = D^{-1/2}WD^{-1/2}$. The following three facts are applied during the transformation above:

1) $\mathbb{W}$ is symmetric, since $W$ is symmetric.
2) $\mathbb{D}_{\alpha\alpha} = \sum_{\beta=1}^{N^2} \mathbb{W}_{\alpha\beta}$, since

$$\begin{aligned}
\mathbb{D}_{\alpha\alpha} = D_{ii}D_{kk} &= \sum_{j=1}^{N} W_{ij}\sum_{l=1}^{N} W_{kl} \\
&= \sum_{j=1}^{N}\sum_{l=1}^{N} W_{ij}W_{kl} = \sum_{\beta=1}^{N^2} \mathbb{W}_{\alpha\beta}.
\end{aligned} \tag{3}$$

3) $\mathbb{S} = \mathbb{D}^{-1/2}\mathbb{W}\mathbb{D}^{-1/2}$, since

$$\begin{aligned}
\mathbb{S}_{\alpha\beta} &= S_{ij}S_{kl} \\
&= D_{ii}^{-1/2}W_{ij}D_{jj}^{-1/2}D_{kk}^{-1/2}W_{kl}D_{ll}^{-1/2} \\
&= D_{ii}^{-1/2}D_{kk}^{-1/2}W_{ij}W_{kl}D_{jj}^{-1/2}D_{ll}^{-1/2} \\
&= \mathbb{D}_{\alpha\alpha}^{-1/2}\mathbb{W}_{\alpha\beta}\mathbb{D}_{\beta\beta}^{-1/2}.
\end{aligned} \tag{4}$$

In summary, the objective function in Eq. (1) is equivalent to

$$J = \vec{A}^{\mathrm{T}}(I-\mathbb{S})\vec{A} + \mu\|\vec{A}-\vec{Y}\|^2. \tag{5}$$

By taking the partial derivative of $J$ with regard to $\vec{A}$, we obtain

$$\frac{\partial J}{\partial \vec{A}} = 2(I-\mathbb{S})\vec{A} + 2\mu(\vec{A}-\vec{Y}). \tag{6}$$

By setting Eq. (6) to zero, we have

$$\vec{A} = \frac{\mu}{\mu+1}\left(I - \frac{1}{\mu+1}\mathbb{S}\right)^{-1}\vec{Y}. \tag{7}$$

After applying $\mathrm{vec}^{-1}$ to both sides of Eq. (7) and setting $\alpha = \frac{1}{\mu+1}$, we obtain

$$\begin{aligned}
A^* &= (1-\alpha)\mathrm{vec}^{-1}\left((I-\alpha S\otimes S)^{-1}\mathrm{vec}(Y)\right), \\
&= (1-\alpha)\mathrm{vec}^{-1}((I-\alpha\mathbb{S})^{-1}\vec{Y}).
\end{aligned} \tag{8}$$

Since Eq. (5) is convex with respect to $\vec{A}$, $A^*$ is the optimal closed-form solution of Eq. (1). As can be clearly seen, the equilibrium state relates to the adjacency matrix $S\otimes S$ of the tensor product graph $\mathbb{G}$, which naturally takes into account the high order relationships between data points.

$\vec{A}$ in Eq. (5) can be deemed as a function, which gives each vertex in $\mathbb{G}$ (also a pair of vertices in the original graph) a real value to describe the pairwise relationship. $I-\mathbb{S}$ is the normalized graph Laplacian of the tensor product graph. So, the proposed method also aims at taking graph Laplacian as a smooth operator to preserve the local manifold structure as [29]. However, the key insight of our approach is utilizing tensor-order graph Laplacian to smooth the pairwise relationship in the original graph.

### 3.2 Iteration-based Solver

Solving RDP using the closed-form solution in Eq. (8) is too computationally demanding (refer to Sec. 4.3 for detailed analysis). To remedy this, we propose an efficient iteration-based solver following [16].

In RDP, an iterative solver can be

$$A^{(t+1)} = \alpha S A^{(t)} S^{\mathrm{T}} + (1-\alpha)Y. \tag{9}$$

To facilitate the iteration, we need to initialize $A^{(1)}$. Opposed to most variants of diffusion process summarized in [1], we do not consider different types of initialization $A^{(1)}$, since our algorithm is guaranteed to converge to the same solution. The only difference is that the convergence speed is not the same with different initializations as demonstrated in the experiments. In each iteration, similarity values are propagated on the affinity graph through the contextual information around both query nodes and database nodes, which is involved by pre-multiplying $A^{(t)}$ by $S$ and post-multiplying $A^{(t)}$ by $S^{\mathrm{T}}$. In other words, the considered context is bidirectional. To summarize, our update scheme during each iteration is to propagate similarities on the affinity graph with probability $\alpha \in (0,1)$ and go back to the initial affinities $Y$ with probability $(1-\alpha)$.

Theorem 1 proves the iteration converges to exactly the same solution presented in Eq. (8) obtained by the regularization framework of RDP. This provides a different yet important explantation of diffusion process on tensor

product graph which well reveals its essence, *i.e.*, before convergence, the iterative similarity propagation is always decreasing the objective value of Eq. (1), in turn, maximizing the smoothness of the manifold in terms of the newly-defined smoothness criterion. Moreover, the generated equilibrium is independent from the initialization of $A^{(1)}$, which supports our previous claim that the initial value of $A^{(1)}$ is irrelevant in our algorithm.

### 3.3 Limit-based Interpretation

In this section, we show that RDP can be also understood as a diffusion process on a tensor product graph.

As is known, a simple realization of diffusion process on an affinity graph can be done by computing powers of the adjacency matrix of the graph. In this paper, the edge weights at time $t$ can be obtained from $(\alpha S)^t$. Many previous works [6], [7], [14] find that it is crucial to stop the diffusion process at a "right" time $t$. However, this is usually problematic especially when no labelled data are available. To remedy this, accumulating the results at different $t$ is suggested [33], [41]. When $t \to \infty$, the limit of the accumulation is

$$\sum_{i=1}^{\infty} (\alpha S)^i = (I - \alpha S)^{-1}. \tag{10}$$

Since the Kronecker product of the adjacency matrix of the graph with itself is the adjacency matrix of tensor product graph, diffusion process on tensor product graph can be simply achieved by replacing $S$ in Eq (10) with $\mathbb{S} = S \otimes S$, thus yielding

$$\mathbb{S}^* = \sum_{i=1}^{\infty} (\alpha \mathbb{S})^i = (I - \alpha \mathbb{S})^{-1}. \tag{11}$$

Note that $\mathbb{S}^* \in \mathbb{R}^{N^2 \times N^2}$, and our aim is to learn a new context-sensitive similarity $A^* \in \mathbb{R}^{N \times N}$. Therefore, we need to gather a portion of elements in $\mathbb{S}^*$ to substitute $A^*$. In this paper, it can be achieved by

$$A^* = \mathrm{vec}^{-1}(\mathbb{S}^* \vec{Y}), \tag{12}$$

where $Y \in \mathbb{R}^{N \times N}$ determines the entry indices of the selected elements in $\mathbb{S}^*$. Meanwhile, since $Y$ does not need to be binary containing only 0 or 1, it also specifies a degree, to which extent the elements in $\mathbb{S}^*$ should be selected.

By multiplying a constant weight $(1-\alpha)$, Eq. (12) is identical to Eq. (8), which suggests that the proposed method is essentially a variant of diffusion process operating on tensor product graph.

### 3.4 Metric-based Interpretation

We present in this section that RDP is tightly related with soft cosine similarity (see Definition 1).

Let $S_i = [S_{i1}, S_{i2}, ..., S_{iN}] \in \mathbb{R}^{1 \times N}$ be the $i$-th row of $S$. Then, the similarity between $x_k$ and $x_l$ associated to the $t$-th propagation step of Eq. (9) can be expressed as

$$< S_k, S_l >= \sum_{i,j=1}^{N} A_{ij}^{(t)} S_{ki} S_{lj}, \tag{13}$$

where we omit the norm of $S_k$ and $S_l$ for approximation.

As $S_k$ (or $S_l$) records the contextual distribution of $x_k$ (or $x_l$), *i.e.*, its neighbors which are visually similar to $x_k$ (or $x_l$). The propagation step of RDP actually computes the soft cosine similarity between two context vectors $S_k$ and $S_l$. The difference is that instead of using a fixed correlation matrix as [42], the correlation matrix $A_{ij}^{(t)}$ is updated dynamically at each iteration. It also implies that it is crucial for diffusion process to possess the property of convergence of iteration and the robustness to the initialization of $A^{(1)}$.

In this sense, RDP is related to those algorithms which leverage the comparison of ranking list or neighborhood to refine the input search results, such as RL-Sim Re-ranking [10], Reciprocal kNN Graph Learning [37], kNN Re-ranking [9], RNN Re-ranking [8]. Most those methods only simply count how many *common* neighbors which $x_k$ and $x_l$ have. However, this strategy may lead to unsatisfactory performances, since it often occurs that two points belong to the same dense cluster, but have no common neighbors. In comparison, the strategy that RDP adopts is not so strict. It considers how many *similar* neighbors which $x_k$ and $x_l$ have, by using the learned correlation matrix $A^{(t)}$.

## 4 VARIANTS

In this section, two important variants of RDP are proposed towards two different goals.

In Sec. 4.1, we extend RDP to tackle two heterogeneous graphs, with each graph from one particular data domain. Since the bidirectional context is asymmetric in this specific scenario, we name this variant as **A**symmetric **R**egularized **D**iffusion **P**rocess (ARDP). By doing so, ARDP can be applied to improve the performance of cross-modal retrieval.

In Sec. 4.2, we generalize RDP to hypergraph, so that more complex relationships (instead of pairwise relationships) between objects can be handled. Since affinity learning here is done by performing **RDP** on the tensor product of the **H**ypergraph with itself, we call this variant as HRDP.

### 4.1 Learning on Heterogeneous Graphs

Assume $\mathcal{G}^{(1)} = \left(X^{(1)}, W^{(1)}\right)$ and $\mathcal{G}^{(2)} = \left(X^{(2)}, W^{(2)}\right)$ are from two heterogeneous data domains, where $X^{(i)}$ denotes $N_i$ data points in the $i$-th domain, and $W^{(i)} \in \mathbb{R}^{N_i \times N_i}$ is the graph adjacency matrix, respectively ($i = 1, 2$). Now, we need to derive the similarity $A \in \mathbb{R}^{N_1 \times N_2}$, which measures the pairwise similarities between the data points in one domain and the data points in the other domain. Accordingly, we can have $D^{(i)} \in \mathbb{R}^{N_i \times N_i}$ and $W^{(i)} \in \mathbb{R}^{N_i \times N_i}$ ($i = 1, 2$). Note $A$ is not necessarily a square matrix, as the sizes of those two domains may be different.

Let $x_k$ and $x_l$ be two exemplars in the first domain, and $x_i$ and $x_j$ be two exemplars in the second domain. The objective function of Asymmetric Regularized Diffusion Process (ARDP) is given as

$$\begin{aligned}\min_A \frac{1}{2} \sum_{i,j=1}^{N_2} \sum_{k,l=1}^{N_1} W_{ij}^{(2)} W_{kl}^{(1)} \left( \frac{A_{ki}}{\sqrt{D_{ii}^{(2)} D_{kk}^{(1)}}} - \frac{A_{lj}}{\sqrt{D_{jj}^{(2)} D_{ll}^{(1)}}} \right)^2 \\ + \mu \sum_{i=1}^{N_2} \sum_{k=1}^{N_1} (A_{ki} - Y_{ki})^2,\end{aligned} \tag{14}$$

where $Y \in \mathbb{R}^{N_1 \times N_2}$ denotes the original cross-modal similarity which we want to preserve, and $\mu$ is the regularization parameter which has the same effect as in the standard RDP.

Defining the two identical coordinate transformations as $\alpha \equiv N_1(i-1)+k$ and $\beta \equiv N_1(j-1)+l$, we can convert Eq. (14) in the same form as presented in Eq. (5), via defining $\mathbb{S} = S^{(2)} \otimes S^{(1)} \in \mathbb{R}^{(N_1N_2)\times(N_1N_2)}$. Also, the closed-form solution $A^* \in \mathbb{R}^{N_1 \times N_2}$ can be adapted from Eq. (8). Moreover, the iterative formulation of ARDP is

$$A^{(t+1)} = \alpha S^{(1)} A^{(t)} {S^{(2)}}^{\mathrm{T}} + (1-\alpha)Y. \tag{15}$$

It is easy to prove that the above iteration converges to the closed-form solution of Eq. (14).

Since RDP is only suitable for within-domain retrieval, one can set $Y = I$ to enforce the self-similarity. However, it is not the case for ARDP, since the data points from different domains are different. From a mathematical point of view, $Y$ is also not a square matrix in general. This indicates that we need to initialize $Y$ by using other algorithms which can provide the cross-modal similarity. In this sense, ARDP can serve as a postprocessing procedure for other cross-modal algorithms to learn more reliable cross-modal similarities. Meanwhile, the acquisition of $S^{(i)}$ is simple and can be done within each individual domain.

### 4.2 Learning on Tensor Product Hypergraph

Both RDP and almost all the aforementioned diffusion processes [1], [4], [14], [15] assume pairwise relationships between objects. To handle more complex relationships, we propose a novel and important variant called HRDP, to show how to perform affinity learning on the tensor product hypergraph in this section.

Apart from a simple graph where each edge connects two vertices, the edge in a hypergraph [43] connects more than two vertices. Let $\mathcal{G} = (X, W)$ denote the hypergraph with $N$ vertices and $M$ hyperedges. The vertices of the graph denote the data points $X = \{x_1, x_2, \ldots, x_N\}$. Each hyperedge $\epsilon$ in $\mathcal{G}$ is assigned a weight $W_{\epsilon\epsilon}$, with all the weights stored in a diagonal matrix $W \in \mathbb{R}^{M\times M}$. The hypergraph $\mathcal{G}$ can be denoted by an incidence matrix $H \in \mathbb{R}^{N\times M}$ as

$$H(i,\epsilon) = \begin{cases} 1, & \text{if } x_i \in \epsilon \\ 0, & \text{if } x_i \notin \epsilon. \end{cases} \tag{16}$$

Based on $H$, the vertex degree of each vertex $x_i \in X$ is

$$D_{ii} = \sum_{\epsilon=1}^{M} W_{\epsilon\epsilon} H_{i\epsilon}, \tag{17}$$

and the edge degree of hyperedge $\epsilon$ is

$$B_{\epsilon\epsilon} = \sum_{i=1}^{N} H_{i\epsilon}. \tag{18}$$

Note that both $D \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{M\times M}$ are also diagonal matrices.

Let $A \in \mathbb{R}^{N\times N}$ be the target similarity matrix learned by HRDP. The objective function of HRDP is

$$\begin{aligned} \min_A \frac{1}{2} \sum_{\epsilon,\varepsilon=1}^{M} \sum_{i,j,k,l=1}^{N} & \frac{W_{\epsilon\epsilon} H_{i\epsilon} H_{j\epsilon}}{B_{\epsilon\epsilon}} \frac{W_{\varepsilon\varepsilon} H_{k\varepsilon} H_{l\varepsilon}}{B_{\varepsilon\varepsilon}} \\ & \left(\frac{A_{ki}}{\sqrt{D_{ii}D_{kk}}} - \frac{A_{lj}}{\sqrt{D_{jj}D_{ll}}}\right)^2 + \mu \sum_{k,i=1}^{N} (A_{ki} - Y_{ki})^2. \end{aligned} \tag{19}$$

The motivation of HRDP is similar to the standard RDP, but differs in the usage of hyperedges which can capture more complex relationships. More specifically, four data points are also involved simultaneously, *i.e.*, $x_i$ and $x_j$ are from the hyperedge $\epsilon$ ($H_{i\epsilon} = H_{j\epsilon} = 1$), and $x_k$ and $x_l$ are from the hyperedge $\varepsilon$ ($H_{k\varepsilon} = H_{l\varepsilon} = 1$). As illustrated in Fig. 2c, the left smoothness term regularizes that if $x_i$ and $x_j$ are connected by the hyperedge $\epsilon$ with edge weight $W_{\epsilon\epsilon}$, and if $x_k$ and $x_l$ are connected by the hyperedge $\varepsilon$ with edge weight $W_{\varepsilon\varepsilon}$, then the learned similarities $A_{ki}$ and $A_{lj}$ should be similar.

The right fitting term also imposes a probability of fastening the initial similarities $Y$ between objects. However, it is usually trivial to obtain such similarities in the hypergraph settings, since the complex relationships between objects are described by the incidence matrix $H$. Hence, $Y$ can be naturally set to an identity matrix $I$ in this situation.

To derive the solution of Eq. (19), we need three identical coordinate transformations, *i.e.*, $\alpha \equiv N(i-1)+k$, $\beta \equiv N(j-1)+l$ and $\gamma \equiv M(\epsilon-1)+\varepsilon$. Then, the smoothness term of Eq. (19) can be described as

$$\begin{aligned} & \frac{1}{2} \sum_{\gamma=1}^{M^2} \sum_{\alpha,\beta=1}^{N^2} \frac{\mathbb{W}_{\gamma\gamma} \mathbb{H}_{\alpha\gamma} \mathbb{H}_{\beta\gamma}}{\mathbb{B}_{\gamma\gamma}} \left(\frac{\vec{A}_\alpha}{\sqrt{\mathbb{D}_{\alpha\alpha}}} - \frac{\vec{A}_\beta}{\sqrt{\mathbb{D}_{\beta\beta}}}\right)^2 \\ &= \sum_{\gamma=1}^{M^2} \sum_{\alpha,\beta=1}^{N^2} \frac{\mathbb{W}_{\gamma\gamma} \mathbb{H}_{\alpha\gamma} \mathbb{H}_{\beta\gamma}}{\mathbb{B}_{\gamma\gamma}} \left(\frac{\vec{A}_\alpha^2}{\mathbb{D}_{\alpha\alpha}} - \frac{\vec{A}_\alpha \vec{A}_\beta}{\sqrt{\mathbb{D}_{\alpha\alpha}\mathbb{D}_{\beta\beta}}}\right) \\ &= \sum_{\gamma=1}^{M^2} \sum_{\alpha=1}^{N^2} \frac{\mathbb{W}_{\gamma\gamma} \mathbb{H}_{\alpha\gamma} \vec{A}_\alpha^2}{\mathbb{D}_{\alpha\alpha}} \sum_{\beta=1}^{N^2} \frac{\mathbb{H}_{\beta\gamma}}{\mathbb{B}_{\gamma\gamma}} - \sum_{\alpha,\beta=1}^{N^2} \sum_{\gamma=1}^{M^2} \vec{A}_\alpha \frac{\mathbb{H}_{\alpha\gamma} \mathbb{W}_{\gamma\gamma} \mathbb{H}_{\beta\gamma}}{\mathbb{B}_{\gamma\gamma}\sqrt{\mathbb{D}_{\alpha\alpha}\mathbb{D}_{\beta\beta}}} \vec{A}_\beta \\ &= \sum_{\alpha=1}^{N^2} \frac{\vec{A}_\alpha^2}{\mathbb{D}_{\alpha\alpha}} \sum_{\gamma=1}^{M^2} \mathbb{W}_{\gamma\gamma} \mathbb{H}_{\alpha\gamma} - \vec{A}^{\mathrm{T}} \mathbb{D}^{-1/2} \mathbb{H} \mathbb{W} \mathbb{B}^{-1} \mathbb{H}^{\mathrm{T}} \mathbb{D}^{-1/2} \vec{A} \\ &= \sum_{\alpha=1}^{N^2} \vec{A}_\alpha^2 - \vec{A}^{\mathrm{T}} \mathbb{D}^{-1/2} \mathbb{H} \mathbb{W} \mathbb{B}^{-1} \mathbb{H}^{\mathrm{T}} \mathbb{D}^{-1/2} \vec{A} \\ &= \vec{A}^{\mathrm{T}} (I - \mathbb{D}^{-1/2} \mathbb{H} \mathbb{W} \mathbb{B}^{-1} \mathbb{H}^{\mathrm{T}} \mathbb{D}^{-1/2}) \vec{A}, \end{aligned} \tag{20}$$

where $\mathbb{W} = W \otimes W \in \mathbb{R}^{M^2\times M^2}$, $\mathbb{H} = H \otimes H \in \mathbb{R}^{N^2\times M^2}$, $\mathbb{B} = B \otimes B \in \mathbb{R}^{M^2\times M^2}$, and $\mathbb{D} = D \otimes D \in \mathbb{R}^{N^2\times N^2}$. The above transformation utilizes the following facts

1) $W_{\epsilon\epsilon} W_{\varepsilon\varepsilon} = \mathbb{W}_{\gamma\gamma}$, $H_{i\epsilon} H_{k\varepsilon} = \mathbb{H}_{\alpha\gamma}$, $H_{j\epsilon} H_{l\varepsilon} = \mathbb{H}_{\beta\gamma}$, $B_{\epsilon\epsilon} B_{\varepsilon\varepsilon} = \mathbb{B}_{\gamma\gamma}$,
2) $\mathbb{B}_{\gamma\gamma} = \sum_{\beta=1}^{N^2} \mathbb{H}_{\beta\gamma}$,
3) $\mathbb{D}_{\alpha\alpha} = \sum_{\gamma=1}^{M^2} \mathbb{W}_{\gamma\gamma} \mathbb{H}_{\alpha\gamma}$.

Afterwards, Eq. (19) can be converted to the same formulation as Eq. (5) with $\mathbb{S} = \mathbb{D}^{-1/2} \mathbb{H} \mathbb{W} \mathbb{B}^{-1} \mathbb{H}^{\mathrm{T}} \mathbb{D}^{-1/2}$. Therefore, the closed-form solution of HRDP can be adapted from Eq. (8). The iterative framework can be also adapted from Eq. (9) via defining $S = D^{-1/2} H W B^{-1} H^{\mathrm{T}} D^{-1/2}$.

### 4.3 Complexity Analysis

In this section, we analyze the algorithmic complexity of the proposed three methods.

Eq. (8) suggests that RDP requires $O(N^4)$ space complexity and $O(N^6)$ time complexity when the closed-form expression is directly used. Such a complexity is impractical even for small graphs. The iterative solver presented in Eq. (9) significantly reduces the complexity, requiring $O(N^2)$ in the space and $O(N^3)$ in the time. Certainly, some mathematical optimization can further reduce the complexity of matrix multiplication. For example, if optimized CW-like algorithms are used, the time complexity of RDP decreases to $O(N^{2.373})$. The complexity of ARDP is slightly different, as it handles two different domains probably in different sizes. However, if we assume the two domains have the same scale, *i.e.*, $O(N_1) = O(N_2) = O(N)$. Its complexity is exactly the same as RDP. As for HRDP, it appears that it incurs much heavier cost than RDP, since six matrices are multiplied to compute the transition matrix $S$. Nevertheless, $D$, $W$ and $B$ all have non-zeros elements only on their diagonal, making the operation computationally cheap. Hence, HRDP also shares the similar complexity with RDP.

Throughout our experiments below, the iterative solver is used in light of complexity.

## 5 Experiments

In this section, we evaluate the validity of the proposed three methods. The experimental comparison of Regularized Diffusion Process (RDP) is given from Sec. 5.1 to Sec. 5.4 with toy problems and real retrieval tasks. ARDP and HRDP are tested in Sec. 5.5 and Sec. 5.6, respectively.

Since the proposed algorithms are guaranteed to converge to the same solution at different initializations of $A^{(1)}$ after a sufficient number of iterations, we set $A^{(1)}$ randomly and the iteration number to 100 if no specified otherwise. In particular for RDP and ARDP, as suggested by [14], it is crucial to constrain diffusion process locally, *i.e.*, only propagating similarities through neighborhood structures. Therefore, graph sparsification is applied by only preserving edges within $k$ nearest neighbors. Since graph sparsification destroys its symmetry, we re-symmetrize it via $W := \frac{W+W^{\mathrm{T}}}{2}$. The regularizer $\mu$ is set to 0.18, indicating that $\alpha \approx 0.85$.

### 5.1 Toy Problems

We first present toy examples to illustrate that RDP can capture well the geometry of manifold structures. The data distribution is a two-spiral pattern as introduced in Fig. 1.

The parameter setup of Manifold Ranking (MR) [17] is the same as RDP, and $Y$ is set to identity matrix $I$. $A^{(1)}$ is set to zero matrix in order to observe the procedure of similarity propagation. In Fig. 3d and Fig. 3h, we present the retrieval results of MR and RDP after convergence (100 iterations). We can find that the retrieval performance of RDP is significantly better than MR. MR fails to reflect the intrinsic structure of two spirals probably because the two spirals are very close.

TABLE 1
The performance comparison with other variants of diffusion process on the ORL, the YALE and the MPEG-7 datasets. The best performances are marked in red and the second best performances are marked in blue.

| Methods | ORL | YALE | MPEG-7 |
|---|---|---|---|
| Baseline | 62.35 | 69.48 | 85.40 |
| SD [4] | 71.67 | 71.46 | 83.09 |
| LCDP [14] | 74.25 | 75.59 | 89.45 |
| TPG [16] | 73.90 | 75.32 | 89.06 |
| MR [17] | 77.05 | 70.85 | 89.26 |
| MR* [17] | 77.58 | 76.91 | 92.61 |
| GDP [1] | 77.42 | 77.30 | 90.96 |
| RDP (Y=I) | 78.53 | 78.07 | 93.77 |
| RDP (Y=W) | 79.27 | 78.24 | 93.78 |

The retrieval results of MR and RDP at different iterations are also given in Fig. 3. Since kNN graph is used, there exist points that do not receive any similarity values at a small amount of iterations, which are marked in gray color. By comparing Fig. 3a with Fig. 3e, Fig. 3b with Fig 3f, and Fig. 3c with Fig 3g respectively, we can observe that RDP exhibits a much faster diffusion speed than MR due to the usage of high order information.

### 5.2 Face and Shape Retrieval

Following the survey paper [1], we then assess the proposed RDP on the ORL face dataset, the YALE face dataset B [20], and the MPEG-7 shape dataset [21].

To ensure a fair comparison, we employ the same parameter setting and the same baselines as in [1]. On two face datasets, $k$ is set to 5 and vectorized raw image pixels are used to represent face images. On MPEG-7 dataset, $k$ is set to 10. However, we do not use AIR descriptor [44], since its performance on the MPEG-7 dataset is already saturated. Instead, we turn to a more frequently-used shape descriptor, Inner Distance Shape Context (IDSC) [45]. The retrieval task is defined as follows: each image is used as query in turn and the rest images serve as the database. The evaluation metric is called Bull's eye score, which counts the recall within top-$K$ returned results. $K = 15$ on two face datasets and $K = 40$ on MPEG-7 dataset.

In Table 1, the comparison with other variants of diffusion process is given, including Self Diffusion (SD) [4], Locally Constrained Diffusion Process (LCDP) [14], Tensor Product Graph (TPG) diffusion [16], Manifold Ranking (MR) [17] and Generic Diffusion Process (GDP) [1]. These baseline methods, except manifold ranking, all use the sparsified affinity graph as RDP. One should first pay attention to the fact that RDP with $Y = W$ achieves almost 1% percent performance boost compared with $Y = I$. The reason behind is that small Euclidean distances are meaningful in retrieval since they can well approximate the small geodesic distances along the manifold. After graph sparsification, $W$ actually only records those small Euclidean distances. Consequently, we can prevent those meaningful relationship from vanishing by setting $Y = W$, thus yielding more reliable performances.

Among the compared methods, LCDP, TPG and GDP can be considered to work on tensor product graph. LCDP cannot guarantee the convergence of iteration. Although

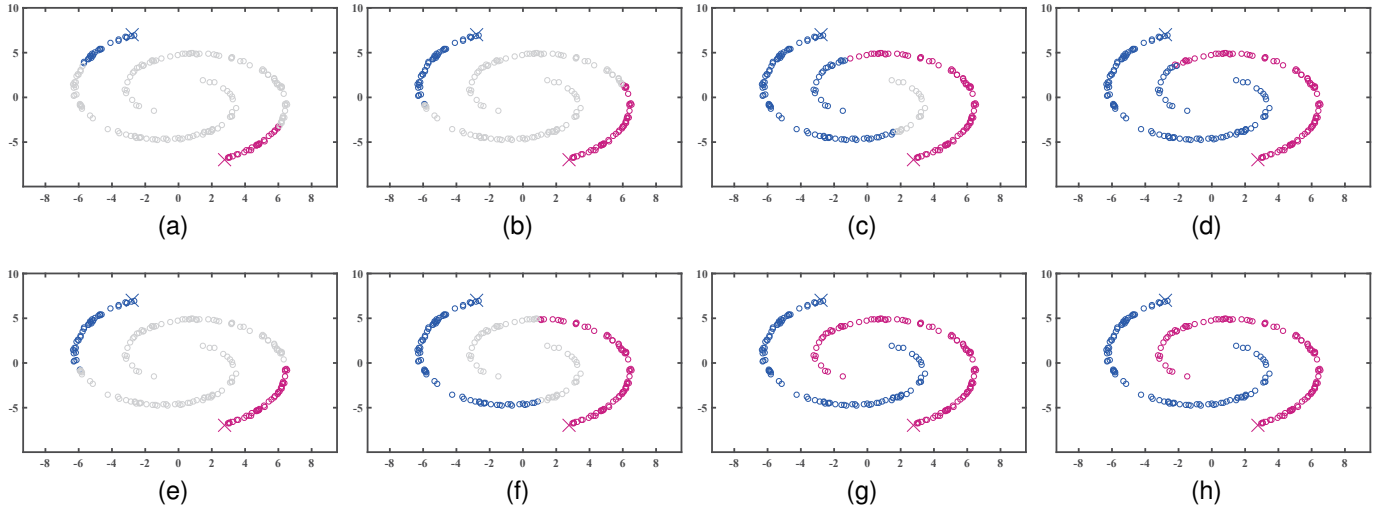


Fig. 3. The two crosses denote the query points. The retrieval results of MR (first row) are given when iteration number is $5$ (a), $10$ (b), $20$ (c) and $100$ (d). The retrieval results of RDP (second row) are given when iteration number is $5$ (e), $10$ (f), $20$ (g) and $100$ (h). The gray points have zero similarities with both query points.

TPG is guaranteed to converge, it lacks a weighting mechanism to balance the contribution of smoothness term and fitting term. As can be seen, RDP outperforms these variants of diffusion process by a large margin. The performance gain is especially valuable, considering that GDP enumerates 72 variants of diffusion process.

Excluding those three diffusion processes, the most related work to ours is MR. Besides the essential difference in the update scheme, the standard MR has three nuances: 1) it spreads affinities on fully-connected graph, while the sparsified graph used by RDP and other methods is proven more robust; 2) it avoids self-reinforcement by setting the diagonal elements of $W$ to zero, while RDP does not; 3) it initializes $Y = I$, while it is demonstrated above that better performances can be achieved with $Y = W$. Hence, we also report the results of a modified version of MR using the three improvements, referred as MR* in Table 1. As the table presents, the modified MR achieves much better performances than its standard version. However, the inferior performances of both two versions of manifold ranking to RDP justify the conclusion that tensor product diffusion is more robust in the scope of object retrieval.

In addition, some other re-ranking algorithms also report the retrieval performances on the MPEG-7 dataset using IDSC as the raw descriptor. Compared with them, RDP is better than Contextual Dissimilarity Measure [2]: $88.30$, Index-Based Re-Ranking [12]: $91.56$, Graph Transduction [3]: $91.61$, RL-Sim Re-Ranking [10]: $92.62$, Mutual kNN Graph [5]: $93.40$, and Sparse Contextual Activation [46]: $93.44$. With the higher baseline ($93.55$ achieved by AIR [44]) used in GDP, the proposed RDP can also yield the perfect performance $100$ as [1].

### 5.3 Natural Image Retrieval

Besides the toy examples and baseline comparisons presented above, the proposed RDP is evaluated with real image retrieval tasks in this section. Four widely-used image datasets are used, including the Ukbench dataset [22], the Holidays dataset [23], the Oxford5K dataset [24] and the Oxford105K dataset.

**Experiments on Ukbench and Holidays.** Ukbench dataset consists of $2,550$ objects, with each object having $4$ different view points. All $10,200$ images are both indexed as queries and database images. The evaluation metric is N-S score, which counts the average recall of the top-4 ranked images. Thus, the best N-S score is $4$. Holidays dataset is composed of $1,491$ images, among which $500$ images serve as the queries. The standard evaluation protocol is mean Average Precision (mAP). The parameter $k$ in RDP is set to $4$.

Driven by the tremendous development of deep learning, the image retrieval performance has been boosted significantly in recent years. To generate the baseline similarity used in RDP, we utilize a representative algorithm [47], which currently achieves the state-of-the-art performances with global image signatures. Using the public available codes, we re-implement its results, *i.e.*, N-S score $3.829$ on the Ukbench dataset and mAP $93.855$ on the Holidays dataset. Note that we use the rotated version of Holidays dataset released in [48]. As can be drawn in Table 2, after applying RDP, its performance is increased to $3.932$ (by $0.103$) on the UKbench dataset and to $95.666$ (by $1.811$) on the Holidays dataset, respectively.

As enormous number of algorithms have reported results on those two datasets, it is unrealistic to give comparisons with all of them. Therefore, besides diffusion processes (*e.g.*, LCMD [15] and TPG [16]), we only include those which are relevant to ours or some other representative algorithms in Table 2. Among them, CDM [2], Graph Fusion [6], [7] and Yang *et al*. [35] are also iteration-based re-ranking algorithms as RDP. kNN Re-ranking [9], RNN Re-ranking [8] and Sparse Contextual Activation (SCA) [46] aim at refining the search result by directly defining a certain metric on the context (without iteration). One may observe that those non-iterative methods usually do not report results on the Holidays dataset. The reason is that most categories on the Holidays dataset only have one groundtruth image. Consequently, it becomes much more difficult to derive a rational similarity by simply counting the number of common neighbors of two images. In comparison, RDP is capable to handle this challenging case due to usage of the correlation

TABLE 2
The performance comparison on the Ukbench and Holidays datasets. The compared methods are sorted by N-S score on the Ukbench dataset in an ascending order. The best performances are marked in red and the second best performances are marked in blue.

| Methods | Ukbench | Holidays |
|---|---|---|
| kNN Re-ranking [9] | 3.56 | - |
| TPG [16] | 3.61 | 68.5 |
| RNN Re-ranking [8] | 3.67 | - |
| CDM [2] | 3.68 | - |
| LCMD [15] | 3.70 | - |
| Graph Fusion [6] | 3.77 | 84.6 |
| Graph Fusion [7] | 3.83 | 84.6 |
| SCA [46] | 3.86 | - |
| Yang *et al*. [35] | 3.86 | 88.3 |
| MSCE [49] | 3.88 | 89.1 |
| Gordo *et al*. [50] | 3.91 | 94.8 |
| RDP (Y=I) | 3.929 | 95.664 |
| RDP (Y=W) | 3.932 | 95.666 |

matrix defined on the context (refer to Sec. 3.4). To support our conjecture, we applied the kNN Re-ranking [9] using the same baseline as RDP, and obtained a poor mAP 0.07. Moreover, we can also observe that the performance difference of RDP between $Y = W$ and $Y = I$ becomes tiny on the two datasets. One possible reason is that there are only a few groundtruth images in each category, making it easier to preserve the local Euclidean distance during the iteration.

Besides, the results of several deep-learning based algorithms have also been reported. In [50], Gordo *et al*. give a thorough extension of [47] (also the baseline used by RDP). By combining query expansion (QE) [18] and database-side feature augmentation (DBA), N-S score on the Ukbench dataset is improved from 3.84 to 3.91, inferior to 3.93 achieved by RDP. QE shares a similar principle with RDP, which uses the features of top-ranked candidates to augment the query feature. On the Holidays dataset, QE is not used as it is not a standard practice. Instead, via extracting descriptors on multiple scales for both query and database images, mAP on the Holidays dataset is improved from 94.0 to 94.8. Multi-scale Contextual Evidences (MSCE) [49] integrates discriminative signatures from the local level, the regional level and the global level via probabilistic analysis, where Convolutional Neural Network (CNN) is used to depict the regional and global patches. Nevertheless, we believe that those descriptors can be also enhanced by RDP.

Fig. 4 gives a qualitative comparison with the baseline for an additional evaluation, which shows RDP can effectively filter false positives at the top of the ranking list.

**Experiments on Oxford5K.** As RDP is graph-based, it has an innate advantage that multiple queries can be concurrently retrieved after the pairwise similarities among the vertices in the graph are updated. However, this also gives RDP a disadvantage that if the query is not the part of the graph, adding new queries to the graph can be time-consuming as the graph-based affinity learning should be done with each query independently.

To handle the scenario where the queries are unseen at the testing time, we get inspirations from a representative algorithm called Regional Diffusion [30] and resort to the following testing procedure: 1) excluding the query images, learn the similarity $A^* \in \mathbb{R}^{N\times N}$ using RDP with $N$ database objects; 2) compute $S_{qj}$, the transition probability from $q$ to all the database objects $j$ ($1 \leq j \leq N$); 3) the similarity between $q$ and a certain database object $i$ after diffusion, that is $A_{qi}$, can be approximated by the weighted average $A^*_{ji}$ over all the $j$, with the weight proportional to $S_{qj}$.

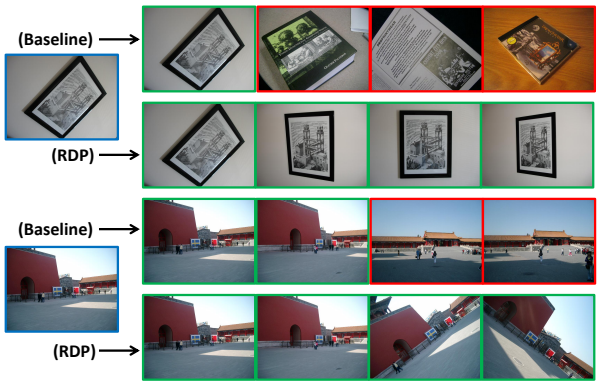


Fig. 4. The qualitative comparison between the baseline and RDP on the Ukbench dataset. Query images are in blue boxes. False positives and true positives are in red and green boxes, respectively.

TABLE 3
The comparison in mAP with the state-of-the-art on the Oxford5K and Oxford105K datasets.

| Methods | Oxford5K | Oxford105K |
|---|---|---|
| Yang *et al*. [35] | 76.2 | - |
| RNN Re-ranking [8] | 81.4 | 76.7 |
| Radenović *et al*. [52] | 85.4 | 82.3 |
| kNN Re-ranking [9] | 88.4 | 86.4 |
| DELF [53] | 90.0 | 88.5 |
| FSR [51] | **95.8** | 93.0 |
| Gordo *et al*. [50] | 94.7 | 93.6 |
| Regional Diffusion [30] | **95.8** | **94.2** |
| RDP | 91.3 | 88.4 |
| QE+DBA+RDP | 95.3 | 94.0 |

The main benefit of such an approximate solution is that we do not need to run RDP for each query. Instead, $A$ is only learned once with database objects, but can be adaptively reused with different queries at the testing time. Namely, we do not add all the queries to the graph when running RDP, but can also index each query one-by-one without graph learning, which makes RDP more flexible and efficient in this specific situation. Readers can refer to [30] for more detailed analysis and a different application where diffusion process is applied to overlapping image regions.

Following [30], [50], the Oxford5K dataset [24] is employed for simulation. The Oxford5K dataset contains $5,062$ images collected from Flickr. Additionally, there are 55 queries, each annotated with a region of interest. Table 3 shows that, the approximate version of RDP can effectively improve the baseline performance of [47] from 86.1 to 91.3. As analyzed above, in [50], a better similarity can be achieved by using QE and DBA additionally. With this baseline, the performance is further improved by RDP from 94.7 to 95.3. It suggests that although the principle of QE and RDP is similar, they also have complementary effects in affinity learning. Meanwhile, this result is comparable to the state-of-the-art, 95.8 reported by fast spectral ranking (FSR) [51] and regional diffusion [30].

**Scalable Experiments on Oxford105K.** To test the potential

capacity in scalable retrieval, RDP is evaluated with the Oxford105K dataset. This dataset is an extension of the Oxford5K dataset, enlarged with $100K$ distractor images from [24].

Since it is computationally prohibitive to directly run graph-based learning algorithms on such large graphs on standard PCs in light of the deficiency in storage and calculating speed, we adopt a truncated solution inspired by query expansion [18] and regional diffusion [30], which runs as follows: 1) for each query, return the top-$500$ ranked images using the baseline similarity; 2) construct the affinity graph with $501$ vertices by including the given query; 3) apply RDP to this smaller graph; 4) refine the ranking list with the learned similarity to the top-$500$ ranked images. As can be drawn in Table 3, the truncated version of RDP still achieves mAP $94.0$ on this challenging dataset, a competitive performance against the state-of-the-art. It is only outperformed by [30], which uses subimages in addition to the whole images. The average indexing time of this truncated solution will be analyzed in Sec. 5.7.

### 5.4 Sketch Retrieval

We also verify RDP with sketch retrieval on the TU Berlin Sketch dataset [25]. It is one of the largest sketch dataset to date, which gathers $20,000$ unique sketches evenly distributed over $250$ object categories.

Similar to N-S score used above, we define the retrieval accuracy as the average recall at top-$80$ ranked sketches, indicating that the best performance is $80$. Each sketch is used as the query in turn, and the rest serves as the database. To represent the sketch, we use the same Bag-of-Words [31] representation built upon a variant of SIFT [54] as described in [25]. Its retrieval accuracy is merely $10.95$. That is to say, among the first $80$ returned candidates, only $11$ are correct positives on average. Obviously, the baseline similarity involves masses of noise in the context, which sets difficulty for RDP to learn similarities. However, our results show that RDP can still learn a better similarity, achieving retrieval accuracy $13.10$ when setting $k = 20$. It firmly demonstrates the robustness of RDP in the presence of considerable noisy context.

### 5.5 Cross-Modal Retrieval

To demonstrate the effectiveness of ARDP in cross-modal retrieval, the Wikipedia dataset [26], [27] is used.

The Wikipedia dataset is generated from Wikipedia's "featured article", a continuously growing collection that has been selected and reviewed by Wikipedia's editors. It contains a total of $2,866$ documents, which are text-image pairs annotated with $10$ semantic categories. The dataset is randomly split into a training set of $2,173$ documents and a test set of $693$ documents. Two retrieval tasks are usually investigated, *i.e.*, image-based text retrieval and text-based image retrieval. In the first case, the images serve as the queries and the texts serve as the database to be indexed. In the second case, the roles of images and texts are reversed. The used evaluation metric is mean Average Precision (mAP), which computes the average precision at the ranks where the recall changes.

TABLE 4
The mAP comparison with the baselines on the Wikipedia dataset. $\surd$ indicates the proposed ARDP is used, while $\times$ indicates not used.

| Baseline | ARDP (ours) | Image | Text | Average |
|---|---|---|---|---|
| CM | $\times$ | 0.249 | 0.196 | 0.223 |
| | $\surd$ | **0.308** | **0.271** | **0.289** |
| SM | $\times$ | 0.225 | 0.223 | 0.224 |
| | $\surd$ | **0.323** | **0.282** | **0.302** |
| SCM | $\times$ | 0.277 | 0.226 | 0.252 |
| | $\surd$ | **0.369** | **0.311** | **0.340** |

TABLE 5
The comparison with the state-of-the-art on the Wikipedia dataset.

| Methods | Image | Text | Average |
|---|---|---|---|
| LCFS [55] | 0.279 | 0.214 | 0.247 |
| LGCFL [56] | 0.279 | 0.217 | 0.248 |
| JFSSL [57] | 0.306 | 0.227 | 0.266 |
| JGRHML [58] | 0.329 | 0.256 | 0.292 |
| ARDP (ours) | **0.369** | **0.311** | **0.340** |

To generate the baseline similarities, the public-available codes and features provided along with the dataset are used. Specifically, each image is represented using an 128-dimensional SIFT [54] histogram in the Bag-of-visual-Words (BoW) [31] model, and each text is represented using a histogram of a 10-topic Latent Dirichlet allocation (LDA) model. Three approaches [26], *i.e.*, Correlation Matching (CM), Semantic Matching (SM), and Semantic Correlation Matching (SCM), are used separately to learn the initial cross-modal similarity $Y$ in Eq. (15). To learn the two within-domain transition matrices $S^{(1)}$ and $S^{(2)}$, the similarity between two images (or texts) is directly obtained by comparing their features in the Euclidean space.

In Table 4, we show the comparison between ARDP and the baseline similarities. As can be drawn, ARDP significantly improves the performances of all the three baselines with both image queries and text queries. For instance, the performances of SCM are improved by $9.20$ percents with image queries, $8.50$ percents with text quires, and $8.80$ percents on average.

Table 5 presents the comparison with other state-of-the-art algorithms, including Learning Coupled Feature Spaces (LFCS) [55], Local Group based Consistent Feature Learning (LGCFL) [56], Joint Feature Selection and Subspace Learning (JFSSL) [57] and Joint Graph Regularized Heterogeneous Metric Learning (JGRHML) [58]. Among them, a closest work to ARDP is JGRHML [58]. Though both use graph regularization, there are many essential differences between JGRHML and ARDP. First, the affinity graph used by ARDP directly models the relationship between data points from two domains, while JGRHML mixes the two domains together to construct a larger affinity graph. Second, ARDP benefits from the similarity constraint of each modality by using the transition matrices within each individual domain. It is known that within-domain similarity is generally more reliable than cross-domain similarity. Third, ARDP enables a simple implementation with only iteration, while JGRHML requires an alternative optimization for multiple variables that needs to compute the matrix inverse. As can be drawn from Table 5, ARDP sets a new state-of-the-art performance on the Wikipedia dataset.

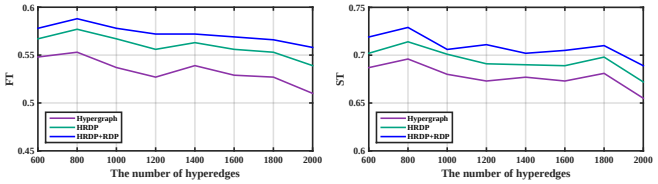


Fig. 5. The performance comparison in FT (a) and ST (b) between hypergraph learning, HRDP and HRDP+RDP on the PSB dataset.

At last, it should be mentioned that ARDP is not limited to text-based image retrieval and image-based text retrieval. It can be expected that ARDP can improve other cross-modal retrieval fields, such as sketch-based 3D shape retrieval [59], [60], sketch-based image retrieval [61], *etc*.

### 5.6 3D Model Retrieval on the Hypergraph

In this section, we assess the performance of HRDP in the hypergraph settings [43] with view-based 3D model retrieval. As a basic and popular methodology in 3D model retrieval, view-based algorithms have drawn much attention for decades. Especially in recent years, the progressive evolution of planar image representation (*e.g.*, BoW [31], convolutional neural network [62]) makes it easier to describe 3D models using depth or silhouette projections.

Following [63], the evaluation pipeline is defined as follows: 1) The views of all 3D models are grouped into multiple clusters via K-means. Each cluster is deemed as one hyperedge that connects the 3D models which have views in this cluster, thus constructing the hypergraph structure $\mathcal{G}$ with the incidence matrix $H$. 2) The weight $W$ of each hyperedge is defined as the sum of the pairwise similarities between any two views in the cluster. 3) The retrieval on the 3D models is performed via running the proposed HRDP on the hypergraph $\mathcal{G}$.

The experiments are conducted on the well-known Princeton Shape Benchmark (PSB) [28], which is comprised of $1,804$ 3D polygonal models. The entire dataset is split into training set and testing set with $907$ models each. Following the convention, only the testing set having $92$ categories is used for evaluation. We employ two evaluation measures, that is First Tier (FT) and Second Tier (ST). The values of FT and ST range from $0$ to $1$, and larger values mean better performances. One can refer to [28] for their mathematical definitions. Following [64], [65], we render $64$ projections for each 3D model. Each projection is fed into the trained CNN model VGG-S [66]. The $L_2$ normalized activations from the 7-th fully connected layer are taken as the view features.

In Fig. 5, we plot the retrieval performances of the baseline (hypergraph learning [43]) and the proposed HRDP, with a different number of the hyperedges. As can been seen clearly, HRDP outperforms the baseline consistently. This firmly demonstrates the positive effects brought by the affinity learning on the tensor product of the hypergraph.

Meanwhile, as expatiated above, the input relationships handled by HRDP are not pairwise as in RDP. It means that though the sophisticated multi-view matching is evaded, we can still obtain the pairwise similarities between 3D models by using HRDP. On the other hand, it also inspires that the output of HRDP can be the input of RDP, since the similarities $A$ learned by HRDP can be naturally converted into the transition matrix $S$ in Eq. (9). Fig. 5 further presents the performances (blue line) of the combined usage of HRDP and RDP. It verifies that HRDP is well compatible with RDP in a cascade manner.

The best performances achieved by our methods are FT $0.588$ and ST $0.729$. Compared with the state-of-the-art, this achievement is higher than tBD [67], 2D/3D Hybrid [68], Makadia *et al*. [69] and PANORAMA [70], but is still inferior to other algorithms, such as 3DVFF [71], GIFT [64], [72], *etc*. However, as emphasized, the focus of this paper is not to establish a developed retrieval system. Instead, we propose a generic affinity learning algorithm. Particularly in this section, we are demonstrating the capacity of HRDP in learning more reliable similarities on the tensor product hypergraph. It can be envisioned that the performance of HRDP can be better if more discriminative features [73], [74], [75] are used.

Of course, HRDP is not restricted by the application of view-based 3D model retrieval. Most tasks which require the affinity learning on the original hypergraph (*e.g.*, clustering) can be reconsidered by using HRDP to bring into account the beneficial high-order (tensor) information. Meanwhile, one could also improve HRDP via feature selection [76], probabilistic modeling, *etc*.

### 5.7 Discussion

**Analysis of Iteration.** In Fig. 6a and Fig. 6e, we present the influence of iteration number on the objective value defined in Eq. (1) and the retrieval performance of RDP on the YALE dataset. Here we set $Y = I$. We use five types of initialization $A^{(1)}$, among which the first $4$ types are used in generic diffusion process [1] and the last one is random values. A first glance at Fig. 6a shows that when propagating affinities on the affinity graph iteratively, RDP tries to minimize the objective function in Eq. (1) until convergence. It reveals the essential behavior of diffusion process on tensor product graph at each iteration. It also means that the number of iteration can be theoretically determined when the objective value reaches its minimum. Second, it is observed that different initializations of $A^{(1)}$ will reach the same equilibrium with different convergence speed. Generally, starting from kNN transition matrix leads to the fastest convergence speed while random initialization is the slowest one. Third, the retrieval performances are exactly the same at equilibrium as presented in Fig. 6e. It demonstrates the robustness of RDP as opposed to the variants summarized in [1] that require a careful initialization of $A^{(1)}$. The same phenomena can be observed on the MPEG-7 dataset, as presented in Fig. 6b and Fig. 6f.

We draw readers' attention that the objective value at the equilibrium is the smallest. However, it does not necessarily indicate that the equilibrium achieves the best retrieval performances. For example, the purple curve of "identity matrix" in Fig. 6e shows that the best performance $79.29\%$ is achieved when iteration number is 7, better than $78.07\%$ after convergence reported in Table 1.

Similar to RDP, we can also observe from Fig. 6c and Fig. 6g that the convergence status of the iteration of ARDP

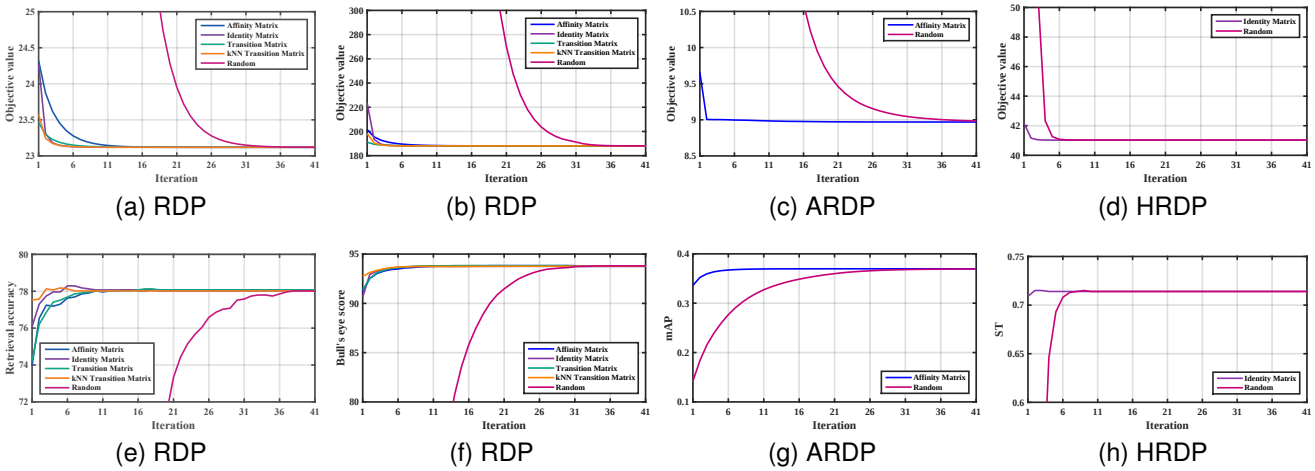


Fig. 6. The objective value (1st row)and the retrieval performance (2nd row) of the proposed approaches as a function of iteration number on the YALE (a)(e), the MPEG-7 (b)(d), the Wikipedia (c)(g), and the PSB (d)(h) datasets.

TABLE 6
The average query time on the 10 datasets used in this work. Note that on the Oxford5K dataset, we use the approximate version of RDP, and on the Oxford105K dataset, we use its truncated version.

| Methods | Datasets | Type | Size | Time |
|---|---|---|---|---|
| RDP | ORL | Face | 400 | $0.51ms$ |
| | YALE | Face | 165 | $0.22ms$ |
| | MPEG-7 | Shape | 1400 | $2.53ms$ |
| | Ukbench | Image | 10,200 | $4.91ms$ |
| | Holidays | Image | 1,491 | $3.11ms$ |
| | Oxford5K | Image | 5,062 | $5.45ms$ |
| | Oxford105K | Image | $100K$ | $809ms$ |
| | TU Berlin | Sketch | $20K$ | $58.9ms$ |
| ARDP | Wikipedia | Text | 2,866 | $0.77ms$ |
| | | Image | 2,866 | $0.80ms$ |
| HRDP | PSB | 3D Model | 907 | $0.54ms$ |

approximates the closed-form solution of its regularization formulation. It also holds for HRDP at Fig. 6d and Fig. 6h.

**Average Query Time.** In Table 6, we present the average query time of the proposed three methods (RDP, ARDP and HRDP) on the 10 datasets used in this work. All the experiments are carried out on a personal computer with an Intel(R) Core(TM) i7 CPU (3.20 GHz) and 64 GB memory.

Owing to the well optimized matrix multiplication, the average query time of the three methods can be controlled within milliseconds on most datasets, even on the relative larger TU Berlin sketch datasets with $20K$ objects. On the Oxford5K dataset, the approximate version of RDP is also efficient to handle the query-by-region case. On the Oxford105K dataset, the average query time of the truncated version of RDP is $809ms$, also acceptable in the scalable setting. In the meantime, it can be expected that the indexing procedure can be significantly accelerated on distributed systems like MapReduce.

**The Impact of Parameters.** There are two key parameters in the proposed RDP, *i.e.*, the regularizer $\mu$ and the number of nearest neighbors on the affinity graph $k$.

In Fig. 7a, we plot the influence of $\mu$ on the retrieval accuracy on the YALE dataset. As it shows, the retrieval performance of RDP changes from 73.88 to 77.07 and arrives the peak value 78.07 at $\mu = 0.18$. As suggested in [1], [14],

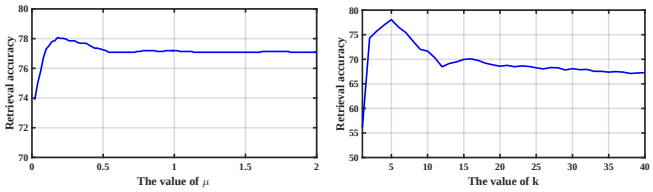


Fig. 7. The influence of the regularizer $\mu$ and the number of nearest neighbors $k$ on the retrieval performance on the YALE dataset.

it is crucial to determine the number of nearest neighbors. Fig. 7b presents that different values of $k$ affects the retrieval performances dramatically. The best performance 78.07 is achieved at $k = 5$, while the worst performance is only 56.14. How to automatically determine the value of $k$ is still an open issue for all approaches that use the pairwise similarity matrices.

## 6 Conclusions

In this paper, we focus on improving object retrieval with diffusion process. Our primary contributions are three tensor-order affinity learning algorithms, customized for different retrieval settings:

1) **RDP** handles simple object retrieval as related works, such as [1], [16]. However, in contrast to those only focusing on the iterative model, the novelty of RDP lies in its regularization framework, which theoretically explains why diffusion process on tensor product graph is more capable in retrieval tasks. Specifically, one can clearly observe that RDP is minimizing a kind of relationship among four tuples at each iteration, so that high order information provided by tensor product graph is necessary.

2) **ARDP** adapts RDP to cross-modal retrieval. The bidirectional context that ARDP imposes to the iterative similarity propagation is derived from two different data domains. Therefore, it can learn the similarity across domains by utilizing the inherent relationship within each individual domain.

3) **HRDP** further generalizes RDP to tackle non-pairwise input relationships. To this end, affinity learning is done on the tensor product of the hypergraph, where the hyperedges are used to capture the complex relationships.

As a result, our work is a generic tool for object retrieval, with the capacity of learning more faithful similarities in most commonly-used retrieval settings. Comprehensive experiments on 10 retrieval benchmarks firmly demonstrate the generalization and the effectiveness of our work. Meanwhile, it can be expected that our work can be a practical guide for other applications, such as geometric verification [18], point registration, graph matching [77], [78] and low-shot learning [19].

## ACKNOWLEDGEMENTS

The code of this work is available at: https://sites.google.com/site/songbaihust/rdp. This work was supported by NSFC 61573160, NSFC 61429201 and NSF IIS-1302164, to Dr. Xiang Bai by the National Program for Support of Top-notch Young Professionals and the Program for HUST Academic Frontier Youth Team, to Dr. Qi Tian by ARO grants W911NF-15-1-0290 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar. The corresponding author of this paper is Xiang Bai.

## REFERENCES

[1] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *CVPR*, 2013, pp. 1320–1327.
[2] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *TPAMI*, vol. 32, no. 1, pp. 2–11, 2010.
[3] X. Bai, X. Yang, L. J. Latecki, W. Liu, and Z. Tu, "Learning context-sensitive shape similarity by graph transduction," *TPAMI*, vol. 32, no. 5, pp. 861–874, 2010.
[4] B. Wang and Z. Tu, "Affinity learning via self-diffusion for image segmentation and clustering," in *CVPR*, 2012, pp. 2312–2319.
[5] P. Kontschieder, M. Donoser, and H. Bischof, "Beyond pairwise shape similarity analysis," in *ACCV*, 2009, pp. 655–666.
[6] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific fusion for image retrieval," in *ECCV*, 2012, pp. 660–673.
[7] ——, "Query specific rank fusion for image retrieval," *TPAMI*, vol. 37, no. 4, pp. 803–815, 2015.
[8] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *CVPR*, 2011, pp. 777–784.
[9] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu, "Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking," in *CVPR*, 2012, pp. 3013–3020.
[10] D. C. G. Pedronette and R. d. S. Torres, "Image re-ranking and rank aggregation based on similarity of ranked lists," *Pattern Recognition*, vol. 46, no. 8, pp. 2350–2360, 2013.
[11] Y. Chen, X. Li, A. Dick, and R. Hill, "Ranking consistency for image matching and object retrieval," *Pattern Recognition*, vol. 47, no. 3, pp. 1349–1360, 2014.
[12] D. C. G. Pedronette, J. Almeida, and R. da Silva Torres, "A scalable re-ranking method for content-based image retrieval," *Information Sciences*, vol. 265, pp. 91–104, 2014.
[13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web." 1999.
[14] X. Yang, S. Koknar-Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval," in *CVPR*, 2009, pp. 357–364.
[15] L. Luo, C. Shen, C. Zhang, and A. van den Hengel, "Shape similarity analysis by self-tuning locally constrained mixed-diffusion," *TMM*, vol. 15, no. 5, pp. 1174–1183, 2013.
[16] X. Yang, L. Prasad, and L. J. Latecki, "Affinity learning with diffusion on tensor product graph," *TPAMI*, vol. 35, no. 1, pp. 28–38, 2013.
[17] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *NIPS*, 2004, pp. 169–176.
[18] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall ii: Query expansion revisited," in *CVPR*, 2011, pp. 889–896.
[19] M. Douze, A. Szlam, B. Hariharan, and H. Jégou, "Low-shot learning with large-scale diffusion," *arXiv preprint arXiv:1706.02332*, 2017.
[20] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *TPAMI*, vol. 23, no. 6, pp. 643–660, 2001.
[21] L. J. Latecki, R. Lakämper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *CVPR*, 2000, pp. 424–429.
[22] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, vol. 2, 2006, pp. 2161–2168.
[23] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, 2008, pp. 304–317.
[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007, pp. 1–8.
[25] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 44–1, 2012.
[26] J. Costa Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. Lanckriet, R. Levy, and N. Vasconcelos, "On the role of correlation and abstraction in cross-modal multimedia retrieval," *TPAMI*, vol. 36, no. 3, pp. 521–535, 2014.
[27] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *ACM international conference on Multimedia*, 2010, pp. 251–260.
[28] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The princeton shape benchmark," in *SMI*, 2004, pp. 167–178.
[29] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NIPS*, 2004, pp. 321–328.
[30] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations," in *CVPR*, 2017.
[31] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, vol. 2, 2005, pp. 524–531.
[32] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process for visual retrieval," in *AAAI*, 2017.
[33] X. Bai, B. Wang, C. Yao, W. Liu, and Z. Tu, "Co-transduction for shape retrieval," *TIP*, vol. 21, no. 5, pp. 2747–2757, 2012.
[34] S. Bai, Z. Zhou, J. Wang, X. Bai, L. Jan Latecki, and Q. Tian, "Ensemble diffusion for retrieval," in *ICCV*, 2017.
[35] F. Yang, B. Matei, and L. S. Davis, "Re-ranking by multi-feature fusion with diffusion for image retrieval," in *WACV*, 2015, pp. 572–579.
[36] S. Zhang, J. Huang, H. Li, and D. N. Metaxas, "Automatic image annotation and retrieval using group sparsity," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 42, no. 3, pp. 838–849, 2012.
[37] D. C. G. Pedronette, O. A. Penatti, and R. d. S. Torres, "Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks," *Image and Vision Computing*, vol. 32, no. 2, pp. 120–130, 2014.
[38] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *TPAMI*, vol. 29, no. 1, 2007.
[39] E. Zemene and M. Pelillo, "Interactive image segmentation using constrained dominant sets," in *ECCV*, 2016, pp. 278–294.
[40] E. Z. Mequanint, Y. T. Tesfaye, H. Idrees, A. Prati, M. Pelillo, and M. Shah, "Large-scale image geo-localization using dominant sets," *TPAMI*, 2018.
[41] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization," *TPAMI*, vol. 28, no. 9, pp. 1393–1403, 2006.
[42] G. Sidorov, A. Gelbukh, H. Gómez-Adorno, and D. Pinto, "Soft similarity and soft cosine measure: Similarity of features in vector space model," *Computación y Sistemas*, vol. 18, no. 3, pp. 491–504, 2014.

[43] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *NIPS*, vol. 19, 2006, pp. 1633–1640.
[44] R. Gopalan, P. Turaga, and R. Chellappa, "Articulation-invariant representation of non-planar shapes," in *ECCV*, 2010, pp. 286–299.
[45] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *TPAMI*, vol. 29, no. 2, pp. 286–299, 2007.
[46] S. Bai and X. Bai, "Sparse contextual activation for efficient visual re-ranking," *TIP*, vol. 25, no. 3, pp. 1056–1069, 2016.
[47] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *ECCV*, 2016, pp. 241–257.
[48] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *ECCV*, 2014, pp. 584–599.
[49] L. Zheng, S. Wang, J. Wang, and Q. Tian, "Accurate image search with multi-scale contextual evidences," *IJCV*, pp. 1–13, 2016.
[50] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *IJCV*, pp. 1–18, 2016.
[51] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Fast spectral ranking for similarity search," *arXiv preprint arXiv:1703.06935*, 2017.
[52] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *ECCV*, 2016, pp. 3–20.
[53] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *ICCV*, 2017.
[54] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[55] K. Wang, R. He, W. Wang, L. Wang, and T. Tan, "Learning coupled feature spaces for cross-modal matching," in *ICCV*, 2013, pp. 2088–2095.
[56] C. Kang, S. Xiang, S. Liao, C. Xu, and C. Pan, "Learning consistent feature representation for cross-modal multimedia retrieval," *TMM*, vol. 17, no. 3, pp. 370–381, 2015.
[57] K. Wang, R. He, L. Wang, W. Wang, and T. Tan, "Joint feature selection and subspace learning for cross-modal retrieval," *TPAMI*, vol. 38, no. 10, pp. 2010–2023, 2016.
[58] X. Zhai, Y. Peng, and J. Xiao, "Heterogeneous metric learning with joint graph regularization for cross-media retrieval." in *AAAI*, 2013.
[59] B. Li, Y. Lu, A. Godil, T. Schreck, B. Bustos, A. Ferreira, T. Furuya, M. J. Fonseca, H. Johan, T. Matsuda *et al.*, "A comparison of methods for sketch-based 3d shape retrieval," *CVIU*, vol. 119, pp. 57–80, 2014.
[60] F. Zhu, J. Xie, and Y. Fang, "Learning cross-domain neural networks for sketch-based 3d shape retrieval," in *AAAI*, 2016, pp. 3683–3689.
[61] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *TVCG*, vol. 17, no. 11, pp. 1624–1636, 2011.
[62] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.
[63] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, "3-d object retrieval and recognition with hypergraph analysis," *TIP*, vol. 21, no. 9, pp. 4290–4303, 2012.
[64] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. Jan Latecki, "Gift: A real-time and scalable 3d shape search engine," in *CVPR*, 2016, pp. 5023–5032.
[65] S. Bai, X. Bai, W. Liu, and F. Roli, "Neural shape codes for 3d model retrieval," *PRL*, vol. 65, pp. 15–21, 2015.
[66] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.
[67] M. Liu, B. C. Vemuri, S. ichi Amari, and F. Nielsen, "Shape retrieval using hierarchical total bregman soft clustering," *TPAMI*, vol. 34, no. 12, pp. 2407–2419, 2012.
[68] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, and S. J. Perantonis, "3d object retrieval using an efficient and compact hybrid shape descriptor," in *3DOR*, 2008, pp. 9–16.
[69] A. Makadia and K. Daniilidis, "Spherical correlation of visual representations for 3d model retrieval," *IJCV*, vol. 89, no. 2, pp. 193–210, 2010.
[70] 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*, vol. 89, no. 2-3, pp. 177–192, 2010.
[71] T. Furuya and R. Ohbuchi, "Fusing multiple features for shape-based 3d model retrieval," in *BMVC*, 2014.
[72] S. Bai, X. Bai, Z. Zhou, Z. Zhang, Q. Tian, and L. J. Latecki, "Gift: Towards scalable 3d shape retrieval," *TMM*, vol. 19, no. 6, pp. 1257–1271, 2017.
[73] 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*, vol. 33, no. 4, pp. 852–858, 2011.
[74] H. Tabia, H. Laga, D. Picard, and P.-H. Gosselin, "Covariance descriptors for 3d shape matching and retrieval," in *CVPR*, 2014, pp. 4185–4192.
[75] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *ICCV*, 2015, pp. 945–953.
[76] Z. Zhang, L. Bai, Y. Liang, and E. Hancock, "Joint hypergraph learning and sparse regression for feature selection," *Pattern Recognition*, vol. 63, pp. 291–309, 2017.
[77] T. Wang, H. Ling, C. Lang, and J. Wu, "Branching path following for graph matching," in *ECCV*, 2016, pp. 508–523.
[78] X. Shi, H. Ling, W. Hu, J. Xing, and Y. Zhang, "Tensor power iteration for multi-graph matching," in *CVPR*, 2016, pp. 5062–5070.

**Song Bai** received the B.S. and Ph.D. degree in Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China in 2013 and 2018, respectively. His research interests include image retrieval and classification, 3D shape recognition, person re-identification and deep learning. More information can be found in his homepage: https://sites.google.com/site/songbaihust/.

**Xiang Bai** received the B.S., M.S. and Ph.D. degrees from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively, all in Electronics and Information Engineering. He is currently a Professor with the School of Electronic Information and Communications, HUST. He is also the Vice-Director of National Center of Anti-Counterfeiting Technology, HUST. His research interests include object recognition, shape analysis, scene text recognition and intelligent systems. He is a senior member of IEEE.

**Qi Tian** received the B.E. degree in electronic engineering from Tsinghua University, China, the M.S. degree in electrical and computer engineering from Drexel University, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 1992, 1996, and 2002, respectively. He is currently a Professor with the Department of Computer Science, University of Texas at San Antonio (UTSA). His research interests include multimedia and computer vision. He is a Fellow of IEEE.

**Longin Jan Latecki** is a professor at Temple University. His main research interests include computer vision and pattern recognition. He has published 250 research papers and books. He is an editorial board member of Pattern Recognition and Computer Vision and Image Understanding. He received the annual Pattern Recognition Society Award together with Azriel Rosenfeld for the best article published in the journal Pattern Recognition in 1998. He is a senior member of the IEEE.