# Understanding Image Retrieval Re-Ranking: A Graph Neural Network Perspective

XUANMENG ZHANG, University of Technology Sydney, NSW, Australia
MINYUE JIANG, Baidu Inc., Beijing, China
ZHEDONG ZHENG, University of Macau, Macau, China
XIAO TAN, Baidu Inc., Beijing, China
YI YANG, Zhejiang University, Zhejiang, China

As a prevailing post-processing tool for image retrieval, re-ranking typically leverages high-confidence retrieved samples to refine the initial ranking list. However, a significant challenge persists: existing re-ranking methods are computationally intensive, leading to impractical time costs for real-world applications. To overcome this limitation, we first review current re-ranking techniques and then introduce a novel real-time approach by reformulating the re-ranking process as a highly-parallel Graph Neural Network (GNN). Specifically, we decompose the traditional neighbor-based re-ranking into two distinct stages: retrieving high-quality gallery samples and refining neighbor similarity. We propose that the first stage can be effectively replaced by constructing a k-nearest neighbor (knn) graph, while the second stage can be implemented by propagating messages within this graph. In practice, since the knn graph is sparse, the GNN only considers a limited number of vertices and their connected edges, enabling efficient updates of vertex features. We validate the effectiveness and efficiency of our approach through extensive experiments on five datasets. For example, on the Market-1501 dataset, our method accelerates the re-ranking process to 9.4ms using a single K40m GPU. Additionally, we observe similar acceleration results on the other four retrieval benchmarks, Paris-6k, Oxford-5k, VeRi-776, and University-1652, while maintaining competitive performance.

CCS Concepts: • **Computing methodologies** → **Search with partial observations**.

Additional Key Words and Phrases: Image retrieval, Graph Neural Networks.

**ACM Reference Format:**
Xuanmeng Zhang, Minyue Jiang, Zhedong Zheng, Xiao Tan, and Yi Yang. 2025. Understanding Image Retrieval Re-Ranking: A Graph Neural Network Perspective. *ACM Trans. Multimedia Comput. Commun. Appl.* 9, 4 (March 2025), 19 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

Re-ranking leverages retrieved samples with high confidence to rank the initial ranking list again [5, 35, 72] (see Fig. 1), which is usually viewed as a post-processing tool. It has been widely

Xuanmeng Zhang is with the School of Computer science, NSW, Australia 2007. E-mail: zhangxuanmeng.zxm@gmail.com. Zhedong Zheng is with the Faculty of Science and Technology, and Institute of Collaborative Innovation, University of Macau, China 999078. E-mail: zhedongzheng@um.edu.mo. Minyue Jiang and Xiao Tan are with Baidu Inc., Beijing, China 311121. E-mail: jiangminyue@baidu.com, tanxiao01@baidu.com.
Yi Yang is with the College of Computer Science and Technology, Zhejiang University, China 310027.
* Zhedong Zheng is the corresponding author.
Authors' addresses: Xuanmeng Zhang, University of Technology Sydney, 2007, NSW, Australia; Minyue Jiang, Baidu Inc., 100193, Beijing, China; Zhedong Zheng, University of Macau, Macau, China; Xiao Tan, Baidu Inc., 100193, Beijing, China; Yi Yang, Zhejiang University, 310013, Zhejiang, China.

ACM 1551-6857/2025/3-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

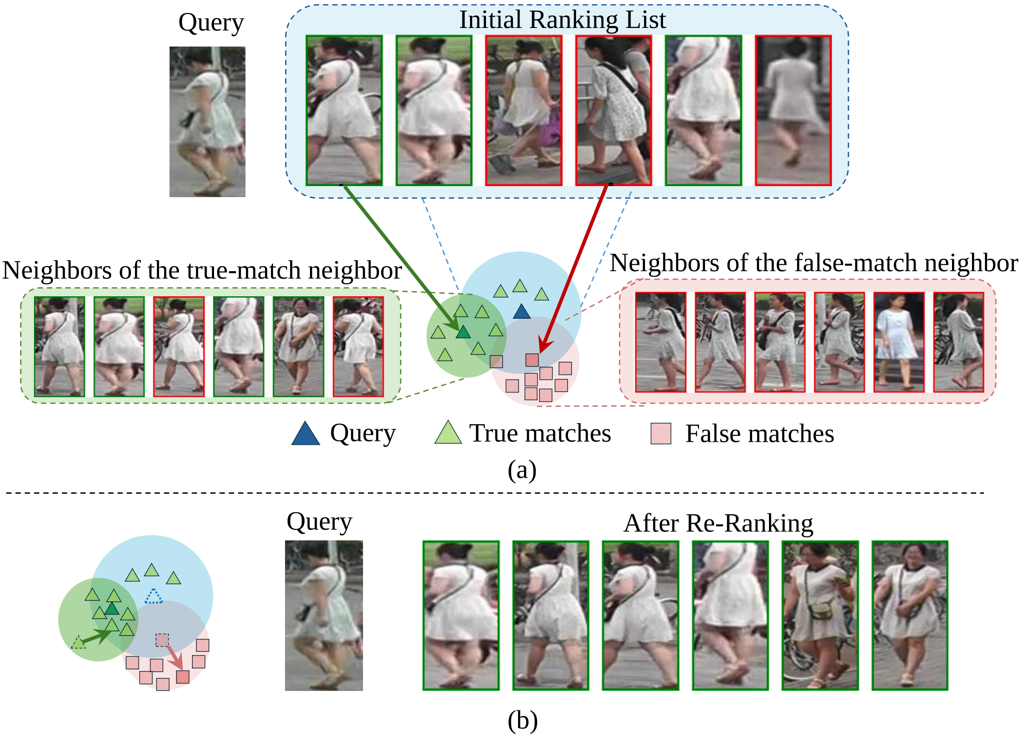

Fig. 1. **Illustration of re-ranking.** The images in green boxes are true matches, and the images in pink boxes are false matches. From subfigure (a), we observe that the negative sample usually attracts neighbors of different identities, while the neighbour of true matches are closer to the original query. Therefore, by comparing the neighbor similarity, re-ranking can filter out the hard negatives from high-confidence retrieved samples and rank the initial ranking list again (see subfigure (b)).

adopted in various of retrieval tasks [39], such as person retrieval [34], vehicle retrieval [25] and localization [67]. Re-ranking methods can be divided into two categories according to similarity criteria, *i.e.*, feature similarity [36] and neighbor similarity [1, 71]. Given a pair of images, feature similarity is generally evaluated based on the Euclidean distance or cosine distance in the feature space. In contrast, neighbor similarity measures the number of shared neighbors. If two samples share more neighbors, they will obtain a higher neighbor similarity score. Generally, the neighbor-based methods outperform the feature-based methods because they are robust to hard negatives, which usually present a different neighbor set compared with the true-matches (see Fig. 1). In this paper, we mainly study the neighbor-based re-ranking methods due to their robustness and superior performance. However, there remains a common and challenging problem that they are time-consuming due to the high computational complexity. For instance, the $k$-reciprocal re-ranking [71] adopts a reciprocal-neighbor rule to calculate the Jaccard distance in a sequential manner, which requires massive set comparison operations.

To address the problem of unaffordable high complexity, we consider the feasibility of the parallel inference. Graph neural network (GNN) is a powerful tool that can process data with topology structures in a parallel way. To leverage the effectiveness and efficiency of graph neural network (GNN), we reformulate the re-ranking process as a high-parallelism GNN to efficiently conduct the re-ranking operations, *e.g.*, neighbor calculation and query expansion. Different from the convolution operation in CNN that computation can only take place within local regions, GNN propagates the features of vertexes and edges over the entire graph. In particular, the neighbor-based re-ranking process can be divided into two phases, *i.e.*, retrieving high-confidence gallery samples and refining neighbor similarity. In the first phase, these methods select the high-confidence samples according to the initial rank list. For the second phase, local query expansion is used to refine the neighbor similarity. We argue that the first stage can be replaced by building the $k$-nearest neighbor ($k$-NN) graph, capturing the topology structure among data. The second phase can be implemented by propagating the message in the graph. Considering the graph is sparse and GNN only concerns vertexes with the connected edges, we can efficiently update vertex features. On the Market-1501 dataset [64], we accelerate the re-ranking processing to **9.4ms** on GPU, facilitating the real-time post-processing for retrieval tasks. Furthermore, we observe similar acceleration results on other benchmarks while maintaining competitive performance. Overall, the main contributions of this work are summarized as follows:

- We identify the challenging problem in applying re-ranking approaches to the real-world scenarios, *i.e.*, large time cost due to high computation complexity. To address this limitation, we revisit re-ranking methods and demonstrate that the re-ranking process can be re-formulated as a high-parallelism Graph Neural Network (GNN), which facilitates the real-time post-processing for retrieval tasks.
- Extensive experiments on five datasets, *i.e.*, Market-1501 [64], VeRi-776 [26], Oxford-5k [32], Paris-6k [33] and University-1652 [67], have shown that the proposed method can significantly accelerate the re-ranking process. Compared to conventional re-ranking methods, our approach achieves competitive or even superior retrieval performance. Additionally, we will release our code as open-source to benefit the research community.

## 2 RELATED WORK

**Image Retrieval.** General image retrieval aims at retrieving an instance of interest across multiple different cameras from different viewpoints. With the urgent demand for public safety in real-world scenarios, there has aroused significant interest in the computer vision community. Retrieval is a challenging task due to the violent viewpoints, image blur, severe occlusions, and varied illumination. Early research works mainly focus on building discriminative handcrafted features [28]. In recent years, with the advancement of deep neural networks, retrieval has achieved significant performance in many benchmarks. Several works design effective loss functions, *e.g.*, triplet loss [12], N-pair loss [40], instance loss [68], contrastive loss [49] and ranked list loss [50], in deep metric learning methods to pull images of the same identity close and push different instances far away [11, 65]. For instance, Zheng *et al.* [69] combine the cross-entropy loss with widely-used label smoothing [29] to optimize the model. Characterized by a siamese network, some re-id approaches [44] are also trained over multiple metric learning objectives. Another popular pipeline for retrieval tasks is to leverage part-level [4] and multi-level [52] fine-grained features. To obtain better matching results, Cheng *et al.* [3] propose a robust fusion strategy by gathering the strengths of various methods, while several works [48, 70] focus on the local part matching. The fine-grained parts can be roughly estimated with prior position knowledge [66], keypoint detection [22], and semantic parsing [16]. The auxiliary information, *e.g.*, attributes [23] and text descriptions [59], further promotes performance by eliminating hard negatives.

**Re-ranking.** Different from model pruning [51] during training time, the re-ranking method is one of the common post-processing acceleration methods for retrieval, which can be divided into two categories according to similarity criteria, *i.e.*, feature similarity, and neighbor similarity. Feature similarity-based methods enrich the query feature and re-calculate the one-to-one similarity. For instance, both Tan *et al.* [41] and Zhang *et al.* [63] introduce a contrastive transformer to re-rank the top-100 matches, while Lee *et al.* [18] further leverage multi-layer features. Wang *et al.* [46] take advantage of query augmentation to conduct similarity ensemble. The average query expansion(AQE) [5] directly adopts the averaged feature the top-$k$ gallery as the new query feature. Radenovic *et al.* [36] propose $\alpha$-weighted query expansion by assigning different weights to the gallery samples. Iscen *et al.* [14] introduce a new retrieval approach by capturing distinct representations in the feature space. The proposed regional diffusion mechanism can handle one or more query features at the same cost. Sparse contextual activation (SCA) [1] is proposed to encode the local neighbor contextual distribution of features with the generalized Jaccard metric. Zhong *et al.* [71] design an effective $k$-reciprocal re-ranking method, which incorporates the advantages of $k$-reciprocal nearest neighbors [35] into SCA [1]. Shen *et al.* [37] explore $k$-reciprocal relation to refine the subgraph similarity, especially for few-shot learning. Another line of approach requires extra image annotations. These algorithms use human interaction or label supervision. Furthermore, several methods are based on the ensemble of different ranking metrics to refine the final list. For example, multiple distance measure fusion, rank aggregation, and ranking list comparison also have proven to be effective in the retrieval tasks. To accelerate the online image search, Yang *et al.* [57] propose a new diffusion algorithm using a simple linear combination in the $k$-NN search process. Without the need for label information, Dou *et al.* [8] directly train the Graph Diffusion Networks (GRAD-Net) to depict the diffusion process with two loss functions. In general, most re-ranking algorithms pay attention to the ranking performance but neglect the computation efficiency for real-world applications. Different from existing methods, we mainly study the efficiency of re-ranking approaches and intend to simultaneously achieve competitive performance and high computation efficiency.

**Graph Neural Networks.** Graphs [60] are the data structure that models plenty of objects (nodes) and the relationships (edges) [55]. Graph Neural Networks (GNNs) [27] are introduced by [10] to process the structured graph data. Gori *et al.* [10] extend GNNs from recursive neural networks (RNNs) to process graphs without losing the topological information. Specifically, existing graph construction approaches [2] can be divided into two categories, *i.e.*, the graph Laplacian spectrum-based method and the hierarchical domain clustering method. Several models also extend the CNNs to graphs. Kipf *et al.* [17] propose to directly encode the graph structure and features of vertexes by a Graph Convolutional Network (GCN). The ability to leverage message-passing mechanisms enables GCN [54] capture collaborative effects in graph data. Other graph models, such as Graph Attention Network(GAT), EdgeConv, and GraphSAGE, are proposed to tackle the graph tasks. In recent years, GNNs are also applied to the computer vision field, including face clustering, 3D person retrieval, action recognition, and weakly supervised object detection [62]. Some researchers have also applied graph models to the retrieval tasks. For instance, a similarity-guided graph neural network (SGGNN) [38] is proposed to produce the similarity estimation and incorporate the graph building to refine feature representations in one batch. Another popular line is similarity propagation and diffusion such as applying random walk on the nearest neighbor graph [24]. For example, Iscen *et al.* [13] propose a hybrid retrieval technique that leverages both temporal and spectral filtering. Recently, Ji *et al.* [15] introduce a new Context-Aware Graph Convolution Network (CAGCN) for Re-ID tasks. By taking a query image and several candidate gallery samples as inputs, CAGCN [15] can output the relationship between the query and each gallery image directly. To exploit the training data in the learning process, these graph-based methods integrate GNN into both the training and testing processes to utilize the relations between vertexes. However, their performance is not as good as traditional re-ranking algorithms. The major limitation of

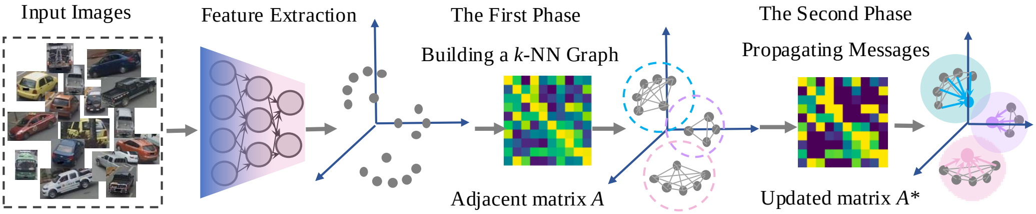

Fig. 2. **The pipeline of our approach**. In the first phase (left), we build the $k$-NN graph and calculate the vertex features, capturing the topology structure among data. For the second phase (right), we employ the GNN to aggregate features from high-confidence samples (vertexes inside the dotted circle). The colored vertexes represent the aggregated features.

this line of methods is that the graph is built on a small set of data points, especially within a mini-batch, which discards massive contextual information in the total dataset. Therefore, the messages propagated by GNN are restricted in a local region and unable to make use of the global information. Similarly, Wang *et al.* [45] also propose Nformer to calculate the neighbor relation via the transformer structure. It requires extra training on the target dataset, and the performance is also largely affected by minority classes. Different from existing methods, we propose a GNN-based re-ranking method building on the entire dataset, capturing the topology structure of the data.

## 3 METHODOLOGY

**Problem Definition.** Given a query image $q$ and a gallery set with $n_g$ images $\mathcal{G} = \{g_i | i = 1, 2, ..., n_g\}$, retrieval is to find relevant images of the query among a large number of candidate images in the gallery. Generally, we map images to a semantic feature space and sort gallery images according to the feature similarity. The re-ranking approach is used to further refine the initial retrieval result.

### 3.1 Neighbor-based Re-ranking

Re-ranking methods usually leverage the extra information based on additional criteria, *i.e.*, the neighbor similarity between images. In this section, we briefly review this typical neighbor-based re-ranking method, which consists of two main steps. In the first step, it encodes the weighted $k$-reciprocal neighbor set, *i.e.*, high-confidence samples, into a $k$-reciprocal feature. In the second step, the $k$-reciprocal feature is improved by the local query expansion, fusing feature representation of neighbor samples. The final distance is calculated as the weighted sum of the original distance and the Jaccard distance.

**Construction of $k$-reciprocal Neighbors.** We define $\mathcal{N}(q, k)$ as the $k$-nearest neighbors (top-$k$ samples in the initial ranking list) of the query image $q$. $\mathcal{R}(q, k)$ is denoted as the $k$-reciprocal nearest neighbors of $q$ in the gallery, which can be formulated as:

$$\mathcal{R}(q, k) = \{g_i | (g_i \in \mathcal{N}(q, k)) \cap (q \in \mathcal{N}(g_i, k))\}. \tag{1}$$

$\mathcal{R}(q, k)$ explicitly considers the neighbors of the nearest samples, enabling the cross-check of neighbor relations. Therefore, the $k$-reciprocal nearest neighbors can effectively reduce the noisy false matches in the high-confidence candidates. **To avoid ambiguity, here we denote the number of the nearest neighbors for $k$-reciprocal feature calculation as $k_1$.** Considering the severe visual appearance changes due to the pose and the occlusion, Zhong *et al.* [71] further introduce a refined expansion set $\mathcal{R}^*(q, k_1)$ by adding the $\frac{1}{2}k_1$-reciprocal nearest neighbors of each candidate in $\mathcal{R}(q, k_1)$ as:

$$\begin{aligned}
&\mathcal{R}^*(q, k_1) \leftarrow \mathcal{R}(q, k_1) \cup \mathcal{R}(g, \frac{1}{2}k_1) \\
&s.t. |\mathcal{R}(q, k_1) \cap \mathcal{R}(g, \frac{1}{2}k_1)| \geq \frac{2}{3}|\mathcal{R}(g, \frac{1}{2}k_1)| \\
&\forall g \in \mathcal{R}(q, k_1),
\end{aligned} \tag{2}$$

where $|\cdot|$ denotes the number of candidates in the set. Given the refined nearest neighbor set $\mathcal{R}^*(q, k_1)$, the neighbor information can be encoded as one $k$-reciprocal feature vector $F_q = [F_{q,g_1}, F_{q,g_2}, ..., F_{q,g_{n_g}}]$, where:

$$F_{q,g_i} = \begin{cases} exp(-d(q, g_i)) & \text{if } g_i \in \mathcal{R}^*(q, k_1) \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

and $d(q, g_i)$ represents the Mahalanobis distance [6] between query image $q$ and gallery image $g_i$. Different neighbors are treated distinctively based on neighbor relations and original pairwise similarities [56].

**Local Query Expansion.** Now we obtain the $k$-reciprocal feature vector for every image. We could apply the local query expansion [5] to further aggregate the similar features within $\mathcal{N}(q, k_2)$. It is worth noting that $k_2$ is different from $k_1$. $k_2$ **represents the number of the nearest $k$-reciprocal neighbors for query expansion.** The final feature $F_q^*$ after the local query expansion can be formulated as:

$$F_q^* = \frac{1}{k_2} \sum_{g_i \in \mathcal{N}(q,k_2)} F_{g_i}. \tag{4}$$

We note that each neighbor is treated equally during the aggregating process. After the expansion of $k$-reciprocal feature, the general Jaccard distance is defined as:

$$d_J(q, g_i) = 1 - \frac{\sum_{j=1}^{n_g} min(F_{q,g_j}^*, F_{g_i,g_j}^*)}{\sum_{j=1}^{n_g} max(F_{q,g_j}^*, F_{g_i,g_j}^*)}. \tag{5}$$

By calculating the ratio of common neighbors and total neighbors of $q$ and $g_i$, the Jaccard distance can measure how dissimilar the two sets are. For example, the distance equals 0 if $q$ and $g_i$ have exactly the same neighbors.

Finally, distance $d^*$ combines the original distance and Jaccard distance to revise the initial ranking list:

$$d^*(q, g_i) = (1 - \lambda)d_J(q, g_i) + \lambda d(q, g_i), \tag{6}$$

where $\lambda \in [0, 1]$ denotes the penalty factor. The refined final distance is subsequently to acquire the re-ranking list.

### 3.2 GNN-based Re-Ranking

In this section, we introduce the GNN-based re-ranking, which reduces the large time cost of complicated operations in neighbor-based re-ranking methods. The key idea is that the similarity between images can be represented as a relation graph. Set comparison operations can be replaced by building a simple but discriminative graph, while features can be updated by the message propagation in GNN. The proposed approach consists of the following two stages. In the first stage, based on the entire image group (query images and gallery images), we construct a graph and encode local information in edges. In the second stage, the proposed GNN propagates messages by aggregating neighbor features with edge weights. The re-ranked retrieval list can be calculated by comparing the similarity of refined vertex features. The pipeline of our approach is shown in Figure 2.

**Graph Construction and Vertex Feature Calculation.** Formally, we denote $X_q$, $X_g$, and $X_u$ as original features of query set, gallery set, and the union of query, and gallery set, respectively. There are $n$ images in the query and gallery set in total. Let $G = (\mathcal{V}, \mathcal{E})$ denote the graph where $\mathcal{V} = \{v_1, ..., v_n\}$ are the vertexes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the edges. Each image is a vertex on the graph and connected edges represent the similarity between vertexes.

First of all, cosine similarity matrix $\boldsymbol{S}$ is calculated:

$$\begin{aligned} &\boldsymbol{S}_{ij} = cos(x_i, x_j), \\ &\text{where } x_i, x_j \in X_u, X_u = X_q \cup X_g. \end{aligned} \tag{7}$$

We can build the $k$-NN graph by connecting the edges of high-confidence samples. The top-$k_2$ samples of each vertex are connected and the weights of edges are defined as:

$$e_{i,j} = \begin{cases} \boldsymbol{S}_{i,j} & \text{if } j \in \mathcal{N}(i, k_2) \\ 0 & \text{otherwise} \end{cases}. \tag{8}$$

Secondly, the feature of every vertex, considering neighbor relations, is calculated. In $k$-reciprocal re-ranking, Zhong *et al.* [71] encode $k$-reciprocal feature by selecting candidates from a expansion of $k$-reciprocal neighbors. Masses of set intersection and comparison operations are required when selecting candidates, leading to huge time costs. We overcome this drawback by adopting a simple but effective strategy to obtain the contextual information of the whole image group. Corresponding to the $k$-reciprocal feature in re-ranking, we aim to encode vertex features by extracting neighbor information from $\boldsymbol{S}$ on the entire graph. Specifically, we first define the adjacent matrix $\boldsymbol{A}$ as:

$$\boldsymbol{A}_{i,j} = \begin{cases} 1 & \text{if } j \in \mathcal{N}(i, k_1) \\ 0 & \text{otherwise} \end{cases}, \tag{9}$$

where $\mathcal{N}(i, k_1)$ is the $k_1$ most similar candidates according to similarity matrix $\boldsymbol{S}$. Generally, an ideal adjacent matrix should be symmetric. Further, the symmetric adjacent matrix $\boldsymbol{A}^*$ is introduced as: $\boldsymbol{A}^* = \frac{\boldsymbol{A}+\boldsymbol{A}^T}{2}$. The value of $\boldsymbol{A}^*_{ij}$ is derived as:

$$\boldsymbol{A}^*_{i,j} = \begin{cases} 1 & \text{if } j \in \mathcal{N}(i, k_1) \wedge i \in \mathcal{N}(j, k_1) \\ 0 & \text{if } j \notin \mathcal{N}(i, k_1) \wedge i \notin \mathcal{N}(j, k_1) \\ 0.5 & \text{otherwise} \end{cases}. \tag{10}$$

We note that $\boldsymbol{A}^*$ encodes more adjacent information. It is because more neighbors are included and adaptive weights are assigned to high-confidence candidates. The experimental results in the ablation study also show $\boldsymbol{A}^*$ performs better than $\boldsymbol{A}$. More importantly, due to the **simplicity**, **symmetry** and **sparsity** of $\boldsymbol{A}^*$, the calculation process can be highly parallelizable and efficient.

Here we define $h_i$ as the feature of vertex $v_i$, which can be extracted from the $i$-th row of the symmetric adjacent matrix $\boldsymbol{A}^*$:

$$h_i = [\boldsymbol{A}^*_{i,1}, ..., \boldsymbol{A}^*_{i,n}]. \tag{11}$$

Such neighbor encoding feature performs better than the original feature because the hard negative images usually share one different neighbor set with the true matches. Hence we construct vertex features based on neighbor similarity rather than directly adopting the original features. The comparison between different features can be found in the ablation study.

**Message Propagation.** In the second phase, a feature aggregating process is required to further improve the retrieval performance, which is achieved by a local query expansion in neighbor-based re-ranking methods. Similarly, in our GNN formulation, this process can be achieved by the message propagating approach [9] on the graph. The key formula of this approach is described as below:

$$h_i^{(l+1)} = h_i^{(l)} + aggregate(\{f(e_{ij}) \cdot h_j^{(l)}\}), \tag{12}$$

where $h_i^{(l)}$ represents the feature of $v_i$ in the $l$-th layer, $f$ is the function to compute the weight of propagating message and *aggregate* represents the aggregator types: *sum*, *mean* or *max*.

We expect to find a suitable function $f_\Theta$, which can capture the relation between vertexes by edge weights. With message propagation, high-confidence vertex features are enhanced and the unreliable vertex features are weakened. Inspired by $\alpha$-weighted query expansion ($\alpha$-QE) [36], we adopt $f_\Theta(e_{ij}) = e_{ij}^{\alpha}$, where $\alpha$ is a fixed value. Then the formula of our modified GNN can be refined as below:

$$\begin{aligned} &h_i^{(l+1)} = h_i^{(l)} + aggregate(\{e_{ij}^{\alpha} \cdot h_j^{(l)}\}), \\ &\text{where} \;\; j \in \mathcal{N}(i, k_2), h_i^{(0)} = h_i \end{aligned} \tag{13}$$

Besides, $h_i^{(l+1)}$ is regularized with $L_2$ norm after every message propagation on the graph. The last GNN layer will output the transformed vertex features $h_i^{(l^*)}$. Finally, we derive the final ranking list according to the cosine similarity of refined features. Since the high-parallelism GNN propagates the message on the sparse graph efficiently, we can update all vertex features simultaneously.

### 3.3 Relation to Existing Methods

Our work is closely related to two primary methodologies in the literature: re-ranking techniques and Graph Neural Networks (GNNs). We provide a novel perspective on the neighbor-based re-ranking process by interpreting it through the lens of GNNs. Specifically, we decompose the traditional neighbor-based re-ranking into two key phases: (1) retrieval of high-quality gallery samples and (2) refinement of neighbor similarity.

*Phase 1 - Retrieval of High-Quality Gallery Samples.* In conventional re-ranking, such as $k$-reciprocal re-ranking [71], the first phase involves identifying the $k$-reciprocal neighbors, which are critical for computing Jaccard distance. This set of high-confidence samples is then expanded by incorporating shared neighbors between the query and candidate images. However, these operations, including the cross-checking and varying number of $k$-reciprocal candidates, are not amenable to hardware acceleration due to their irregular nature and computational complexity In contrast, our method leverages a $k$-Nearest Neighbor ($k$-NN) graph to model relationships, enabling efficient implementation via GPU-friendly matrix operations. This allows for significant speedups and scalability, while maintaining or even improving the quality of the retrieved samples.

*Phase 2 - Refinement of Neighbor Similarity.* The second phase, often referred to as local query expansion, can be viewed as a single-layer GNN with $\alpha = 0$ (in Eq. 13) and the aggregator type sets to *mean*. Conventional methods spread the query expansion message uniformly within a local region, without considering the relative importance of different edges. In contrast, our approach introduces a more sophisticated message-passing mechanism, where vertex features are propagated along with edge weights, effectively integrating both global and local information. This allows for more nuanced and context-aware feature aggregation. Furthermore, our framework supports various types of aggregators, providing flexibility in how information is combined. As shown in our experiments, increasing the number of GNN layers (*e.g.*, two layers) further refines the vertex features, leading to improved performance.

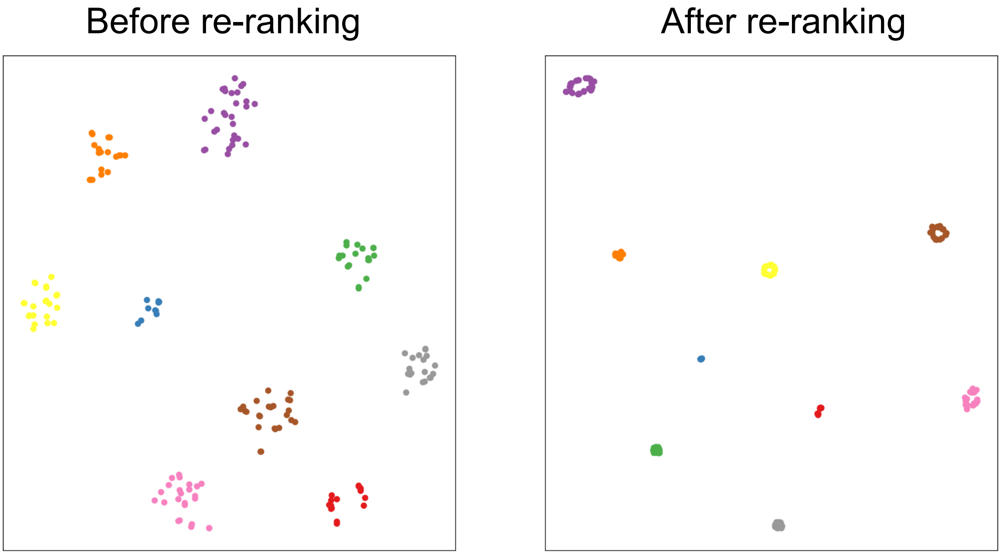

Fig. 3. **Comparison of the feature distribution before and after re-ranking.** We randomly select 9 identities and plot the corresponding features with t-SNE visualization [42]. The different color dots denote different identities. We observe that the feature clusters become more compact after re-ranking.

## 4 EXPERIMENT

We conduct experiments on five retrieval datasets of different application scenarios, including a person retrieval dataset Market-1501 [64], a vehicle retrieval dataset VeRi-776 [26], two landmark retrieval datasets, *i.e.*, Oxford-5k [32] and Paris-6k [33], and a drone-based geo-localization dataset University-1652 [67].

We mainly report mean average precision (mAP) and Recall@K.

**Implementation Details.** When testing, we deploy a two-layer GNN and set the aggregator as *sum* function. The parameter $\alpha$ in Eq. 13 is fixed as 2. For a fair comparison, we adopt the open-source baseline models to verify the effectiveness of the proposed method. In particular, we employ the open-source person retrieval networks[1] as the Market-1501 [64] baseline, which adopts a strong backbone, *i.e.*, ResNet-50-ibn [31], and fuses multi-branch information to enhance the representation. The final feature dimension is 1536. For VeRi-776 [26], we deploy a vehicle retrieval model[2] with the 2048-dimension feature. For Oxford-5k [32] and Paris-6k [33], we extract 2048-dimension feature from the open-source ResNet101-GeM[3] [36]. For University-1652, we follow the official implementation [67] [4] with a 512-dim visual feature. All experiments are conducted on the same machine with one Intel CPU E5-2620 v2 @ 2.10Ghz, 164GB memory, and one K40m GPU with 12GB memory.

[1] https://github.com/douzi0248/Re-ID
[2] https://github.com/BravoLu/open-VehicleReID
[3] https://github.com/filipradenovic/cnnimageretrieval-pytorch
[4] https://github.com/layumi/University1652-Baseline

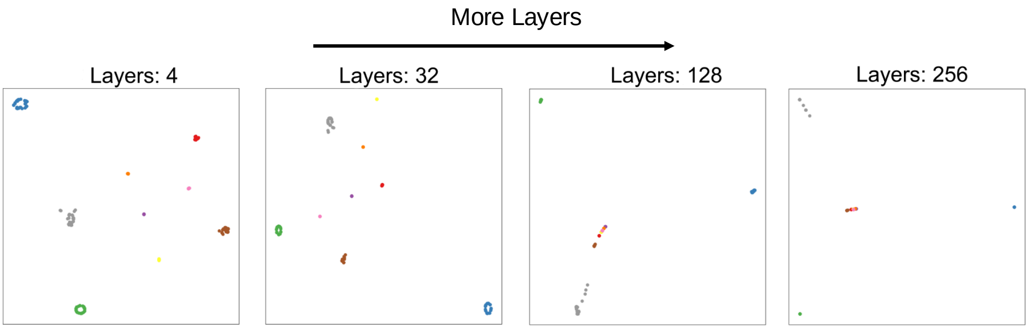

Fig. 4. **t-SNE visualization of the feature distribution.** We use different color dots to represent different identities. When the number of GNN layers reaches 256, we observe features of several different identities converging to the same value, and different clusters become indistinguishable. (Best view when zooming in.)

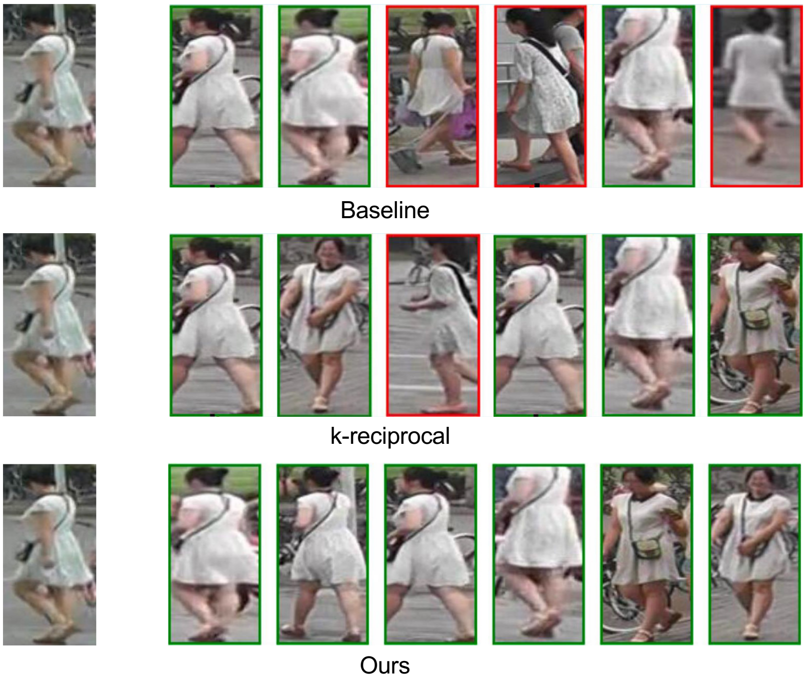

Fig. 5. Top-K retrieval results from (1) the baseline method, (2) the classic k-reciprocal re-ranking method, and (3) our proposed method. The new qualitative results clearly illustrate that our method retrieves more identity-consistent and visually relevant samples compared with the baselines.

### 4.1 Ablation Studies and Further Discussion

**Effect of Different Components.** We study the mechanism of the proposed method on the Market-1501 [64] dataset. We gradually add different components and choose different input vertex features to analyze the contribution of each part. As shown in the first two columns of Table 1, we can see that the symmetric adjacent matrix $\boldsymbol{A}^*$ brings an improvement of 3.07% in

Table 1. **Ablation study.** We study different feature inputs, *i.e.*, original feature $x_i$, adjacent matrix feature $\boldsymbol{A_i}$ and symmetric adjacent matrix feature $\boldsymbol{A_i^*}$. Message propagation means aggregating the feature of neighbor vertexes by Eq 12. Edge weights represent the weights of propagating message in Eq 13. Two-layer GNN denotes the number of layers is 2.

| Components | Performance | | | | | | |
|---|---|---|---|---|---|---|---|
| Vertex features ($x_i$) | ✓ | | | | ✓ | | |
| Vertex features ($\boldsymbol{A_i}$) | | | | | | ✓ | |
| Vertex features ($\boldsymbol{A_i^*}$) | | ✓ | ✓ | ✓ | | | ✓ |
| Message propagation | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Edge weights | | | | ✓ | ✓ | ✓ | ✓ |
| Two-layer GNN | | | | | ✓ | ✓ | ✓ |
| mAP (%) | 88.26 | 91.33 | 94.20 | 94.53 | 93.83 | 94.51 | **94.65** |
| Recall@1 (%) | 95.28 | 93.97 | 95.64 | **96.29** | 95.81 | 95.56 | 96.11 |

mAP from 88.26% to 91.33%. With the help of message propagation, we achieve another 2.87% improvement in mAP and 1.67% improvement in Recall@1 because the relevant samples are pulled much closer. Moreover, we observe that introducing edge weights improve the mAP from 94.20% to 94.53% and the Recall@1 from 95.64% to 96.29%, since vertexes are treated differently according to the similarity. Besides, using two-layer GNN further increases the mAP accuracy by 0.12%. This suggests that aggregating more vertex features appropriately can improve the performance. Finally, we study different input features in the last three columns. It shows that vertex features derived from $\boldsymbol{A}^*$ outperform $\boldsymbol{A}$ and original features $x_i$. It is because the symmetric adjacent matrix feature contains the neighbor similarity instead of the original feature, which is more robust to the hard-negative. To better illustrate the effect of the proposed method, we also provide the t-SNE visualization [42] of the feature distribution before and after re-ranking. We randomly select 9 identities from the Market-1501 dataset and plot the feature embeddings. As illustrated in Fig. 3, we observe that the proposed method produces more compact feature clusters, where the intra-class variance is minimized and the inter-class variance is maximized. We also provide a qualitative case comparison that presents several representative query examples (see Fig. 5). For each query, we show a side-by-side comparison of the Top-K retrieval results from (1) the baseline method, (2) the classic k-reciprocal re-ranking method, and (3) our proposed method. The new qualitative results clearly illustrate that our method retrieves more identity-consistent and visually relevant samples compared with the baselines

**Effect of the GNN Layer Number.** To take one step further, we study the impact of different GNN layer numbers $l^*$ on the Market-1501 dataset. In Figure 6, we report the Recall@1 and mAP with GNN from 1 layer to 256 layers. Over-smoothing is a common phenomenon when stacking more layers into graph neural networks [19, 21]. As illustrated in Fig. 4, we notice that the features of the same identity are gradually gathered with the number of layers increasing. When the number of GNN layers reaches to 256, we observe the features of several different identities converging to the same value, and different clusters become indistinguishable. As a result, the Recall@1 reduces from 96.29% to 93.91%, and the mAP increases from 94.53% to 94.65% then reduces to 93.21%. In Fig 6, the mAP converges to 93.21%, when the GNN layers increase to 256. In this case, the proposed method is approximately equal to the average query expansion (AQE) [5]. Because messages propagate again and again, yielding the over-smoothing result. In contrast, our method has achieved the highest Recall@1 and mAP when using 1 and 2 GNN layers respectively.

**Hyper-parameter Sensitivity.** To analyze the impact of two hyper-parameters ($k_1$ and $k_2$), we conduct experiments on Market-1501 [64] and VeRi-776 [26] datasets. The hyper-parameter $k_1$ is

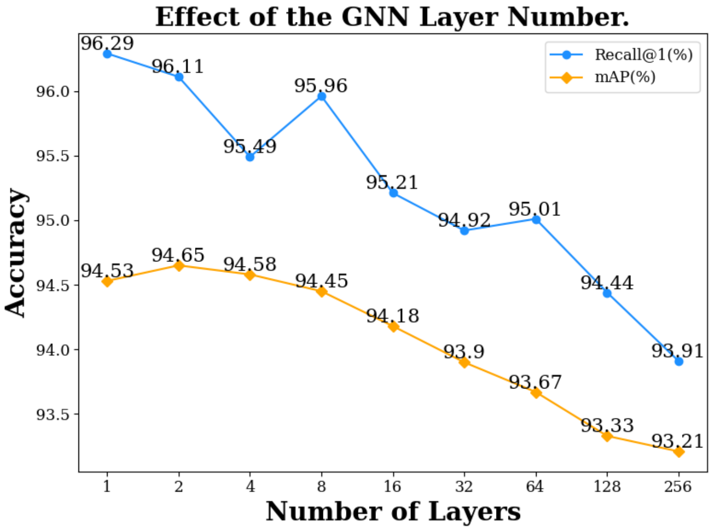

Fig. 6. **Effect of the GNN Layer Number.** We conduct experiments with different number of GNN layers on the Market-1501 [64] dataset. With the number of layers increasing, the mAP accuracy converges to 93.21%. In this case, the proposed method is approximately equal to the average query expansion (AQE) [5].

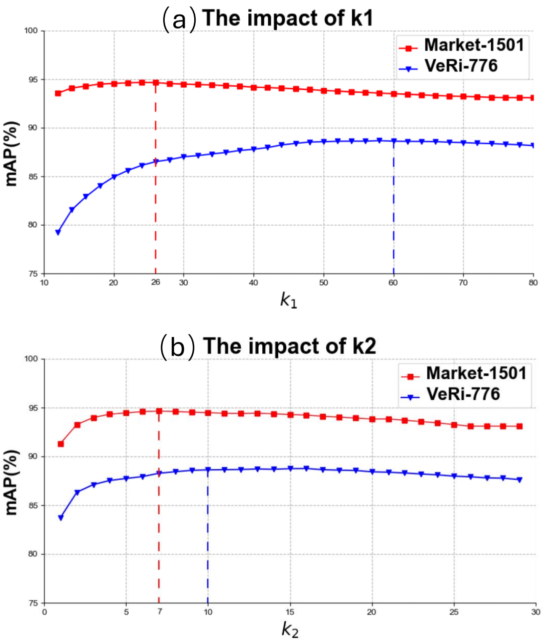

Fig. 7. **The hyper-parameter analysis on $k_1$ and $k_2$.** We analyze hyper-parameters on Market-1501 [64] and VeRi-776 [26]. For Market-1501 [64], following [43], we empirically fix $k_2 = 7$ in **(a)** to study $k_1$, and fix $k_1 = 26$ in **(b)** to study $k_2$. As for VeRi-776 [26], we fix $k_2 = 10$ in **(a)** and $k_1 = 60$ in **(b)** to study the impact of hyper-parameters. (Best view when zooming in.)

Table 2. **Two-phase running time.** We provide the running time of each phase and compare the proposed method with the $k$-reciprocal method. The one-layer GNN is to make a direct comparison with $k$-reciprocal re-ranking, saving 58.6s on CPU. Since our method is GPU-friendly, we also report faster GPU results, while $k$-reciprocal re-ranking is not available on GPU.

| Methods | Platform | Time | | | Performance | |
|---|---|---|---|---|---|---|
| | | Phase 1 | Phase 2 | total | Recall@1 (%) | mAP(%) |
| $k$-reciprocal [71] | CPU | 49.0s | 40.2s | 89.2s | 95.90 | 94.38 |
| **Ours**(1 layer) | CPU | 1.2s | 29.4s | 30.6s | **96.29** | 94.53 |
| **Ours**(2 layers) | CPU | 1.2s | 58.7s | 59.9s | 96.11 | **94.65** |
| **Ours**(2 layers) | GPU | 3.8ms | 5.6ms | 9.4ms | 96.11 | **94.65** |

Table 3. **Time cost.** We compare the proposed method with other post-processing methods on five datasets. Our method finds one balance between speed and performance. For time cost, it is worth noting that the proposed re-ranking method costs the similar average time as SCA. Besides, our method achieves a similar speed with alpha-QE and AQE on GPU, but with a larger accuracy boost (Please see Table 4).

| Methods | Platform | Market-1501 | VeRi-776 | Oxford-5k | Paris-6k | University-1652 | **Average Time** |
|---|---|---|---|---|---|---|---|
| AQE [5] | GPU | 3.4ms | 3.5ms | 1.8ms | 1.7ms | 8.3ms | 3.7ms |
| $\alpha$-QE [36] | GPU | 3.6ms | 3.7ms | 1.9ms | 1.6ms | 9.1ms | 4.0ms |
| Diffusion [57] | GPU | 55.4ms | 30.8ms | 75.2ms | 40.1ms | 69.6ms | 54.2ms |
| SCA [1] | GPU | 568.3ms | 230.7ms | 289.6ms | 278.3ms | 100.3ms | 67.4ms |
| $k$-reciprocal [71] | GPU | 934.9ms | 478.9ms | 712.9ms | 589.3ms | 236.8ms | 99.3ms |
| **Ours** | GPU | 9.4ms | 5.2ms | 3.2ms | 3.5ms | 10.2ms | 6.3ms |
| SCA [1] | CPU | 43.2s | 18.4s | 25.6s | 25.6s | 89.7s | 40.5s |
| $k$-reciprocal [71] | CPU | 89.2s | 36.7s | 64.1s | 64.4s | 135.9s | 78.1s |
| **Ours** | CPU | 60.0s | 21.3s | 17.1s | 23.2s | 85.5s | 41.4s |

used to calculate the adjacent matrix $\boldsymbol{A}^*$, while $k_2$ is the number of neighbors in message passing. Increasing the value of $k_1$ moderately introduces more neighbors. According to the previous work [43], the average number of images per class can be empirically estimated by $[\frac{n}{C}]$ ($C$ represents the number of classes, $[\cdot]$ indicates round down operation) and it is reasonable to construct neighbor relations based on this value, thus we set: $k_1 = [\frac{n}{C}]$. $k_2$ is estimated by a much smaller number than $k_1$ because a large value may bring noisy vertexes. This selection rule is used in all following experiments. We first study the effect of $k_1$ by fixing the value of $k_2$. $k_2$ is set to 7 and 10 on Market-1501 [64] and VeRi-776 [26] respectively. As shown in Fig. 7 (a), our approach is insensitive to $k_1$ in a wide range from 20 to 40 on Market-1501 [64]. For VeRi-776 [26], the proposed method maintains a relatively high performance when $k_1$ changes from 50 to 70. To evaluate the influence of parameter $k_2$, we fix $k_1$ to 26 and 60 on Market-1501 [64] and VeRi-776 [26] separately. Fig. 7 (b) shows the performance of our approach is stable on Market-1501 [64] when $k_2$ changes from 5 to 10. Similar results can be achieved on VeRi-776 [26] when keeping $k_1$ as 60 on VeRi-776 [26] and changing $k_2$ from 10 to 15. In general, the proposed method can achieve comparable results and is relatively robust to a large range of $k_1$ and $k_2$.

**Time Cost Comparison.** We evaluate the efficiency of our method. We analyze the time cost of two phases on the Market-1501 [64] dataset and compare our method with the conventional $k$-reciprocal re-ranking. Here we adopt the official implementation[5]. As shown in Table 2, we test the running time and performance of two methods on the CPU. We can see that both the one-layer and two-layer GNN are faster than $k$-reciprocal re-ranking in terms of the total time cost. To further

[5] https://github.com/zhunzhong07/person-re-ranking

Table 4. **Retrieval performance.** We compare the proposed method with various post-processing methods on five datasets. We observe that the proposed method achieves the best or second-best performance on most datasets.

| Methods | year | Market-1501 | | VeRi-776 | | Oxford-5k | Paris-6k | University-1652 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | mAP (%) | R@1 (%) | mAP (%) | R@1 (%) | mAP (%) | mAP (%) | mAP(%) | R@1 (%) | R@5 (%) | R@10 (%) |
| baseline | - | 88.26 | 95.28 | 78.94 | 95.59 | 88.21 | 92.62 | 63.13 | 58.49 | 78.67 | 85.23 |
| CAGCN [15] | 2020 | 91.70 | 95.90 | 79.60 | 95.80 | - | - | - | - | - | - |
| AQE [5] | 2007 | 93.33 | 95.64 | 82.49 | 89.22 | 90.63 | 96.04 | 71.23 | 67.62 | 83.32 | 86.36 |
| $\alpha$-QE [36] | 2018 | 93.51 | 96.08 | 82.77 | 89.72 | 91.07 | 95.45 | 71.69 | 68.18 | 83.66 | 86.71 |
| Diffusion [57] | 2019 | 93.52 | 95.72 | 86.98 | 95.77 | 92.60 | **97.10** | 72.23 | 70.04 | 85.49 | 86.46 |
| SCA [1] | 2016 | 94.14 | 96.08 | 87.48 | **96.54** | 92.58 | 95.45 | 74.11 | 70.52 | 86.22 | 90.34 |
| $k$-reciprocal [71] | 2017 | 94.38 | 95.90 | 88.44 | 96.36 | 92.31 | 96.49 | 73.67 | **70.71** | 83.86 | 85.65 |
| MGK [53] | 2023 | 92.89 | 78.13 | - | - | - | - | - | - | - | - |
| Cheb-GR [58] | 2025 | 93.00 | 96.00 | - | - | - | - | - | - | - | - |
| Pose2ID [61] | 2025 | 93.01 | 95.52 | - | - | - | - | - | - | - | - |
| FusionTexReIDNet [30] | 2025 | 93.50 | **98.50** | - | - | - | - | - | - | - | - |
| **Ours** | - | **94.65** | 96.11 | **88.61** | 96.42 | **92.95** | 96.21 | **74.11** | 70.30 | **87.53** | **91.21** |

reduce the time cost, we extend the GPU-version of our method, which can accelerate the re-ranking process to **9.4ms**. To the best of our knowledge, there is no available GPU implementation of $k$-reciprocal re-ranking due to the sequential operations. Therefore, we do not include the GPU comparison with [71] but other methods [5, 36, 57]. As shown in Table 3 and Table 4, the proposed method finds one balance point between speed and performance. In general, our method achieves better performance on most datasets. In terms of the time cost on CPU, it is worth noting that the proposed re-ranking method achieves a similar speed to SCA [1]. As for the GPU version, our method achieves a competitive speed with the vanilla methods including alpha-QE [36] and AQE [5], while achieving a better performance improvement.

### 4.2 Retrieval Performance

**Experiments on Market-1501 and VeRi-776.** As shown in Table 4, we compare the proposed method with other post-processing approaches on the two retrieval datasets, *i.e.*, Market-1501 [64] and VeRi-776 [26]. There are two main observations. On the one hand, our method can improve mAP and Recall@1 by a large margin on the baseline. Specifically, our approach increases mAP by **6.39%** and Recall@1 by **0.83%** on Market-1501 [64]. On VeRi-776 [26], we observe that the proposed approach gains 9.67% improvement on mAP and 0.83% improvement on the Recall@1. On the other hand, we compare our approach with a variety of post-processing methods, including four feature similarity-based methods: AQE [5], $\alpha$-QE [36], CAGCN [15], and Diffusion [57], two neighbor similar-based methods: SCA [1] and $k$-reciprocal [71]. Results show that our approach outperforms all other methods both in mAP (**94.65%**) and Recall@1 (**96.11%**) on Market-1501 [64]. As for VeRi-776 [26], our approach has achieved the highest mAP of **88.61%** and second-best Recall@1 of 96.42%. Besides, we also compare the time cost of different post-processing methods both on CPU and GPU platforms in Table 3. The AQE [5], $\alpha$-QE [36], CAGCN [15], and Diffusion [57] are implemented on GPU, while SCA [1] and $k$-reciprocal re-ranking [71] operate on CPU due to the restriction of complex operations. The proposed method only takes **9.4ms** and **5.2ms** to update Market-1501 [64] and VeRi-776 [26] respectively, which is significantly better than traditional neighbor-based re-ranking methods, and competitive to methods based on feature similarity.

**Experiments on Oxford-5k and Paris-6k.** The proposed method is further evaluated on two small-scale landmark retrieval datasets, *i.e.*, Oxford-5k [32] and Paris-6k [33]. The performances of different post-processing methods on ResNet101-GeM [36] are reported in Table 4. The feature dimension is 2048. Our approach has achieved the highest mAP accuracy of **92.95%** on Oxford-5k [32], and competitive results 96.21% mAP accuracy on Paris-6k [33]. The experiment verifies the scalability of the method on small-scale datasets. As shown in Table 3, the re-ranking process only consumes **5.2ms**, which is faster than other re-ranking methods, *i.e.*, SCA, and k-reciprocal, and achieves a similar speed with the vanilla query expansion methods.

**Experiments on University-1652.** We also provide experimental results on the drone-based geo-localization dataset [67]. Given one drone-view image, the drone-view target localization task aims to find the corresponding satellite-view image to localize the target building on the satellite platform. From Table 4, we observe that our method can increase Recall@1 by 11.81%, Recall@5 by 8.86%, Recall@10 by 5.98%, and mAP by 10.98%. Comparing to other methods, our approach achieves the highest performance on Recall@5, Recall@10 and mAP with **87.53%**, **91.21%**, and **74.11%** respectively. From Table 3, we can see that the proposed method only consumes **10.2ms** on GPU and 85.5s on CPU.

## 5 DISCUSSION

### 5.1 Message passing

In graph-network–based re-ranking, information propagation (message passing) is the mechanism that turns pairwise similarities into a globally consistent ranking. Concretely, it lets the score of each candidate be influenced not only by its direct similarity to the query, but also by the structure of its neighborhood (and higher-order neighborhoods) in the similarity graph.

The Key roles it plays are as follows.

(1) **Neighborhood/context aggregation (k-reciprocal and manifold effects).** Each node updates its representation/score by aggregating signals from connected nodes (top-$k$ neighbors). This captures higher-order similarity: two items can become more relevant if they are supported by many shared neighbors, even if their direct query similarity is modest.
(2) **Noise suppression and robustness.** Spurious high-similarity matches (false positives) tend to have weak support from the local graph structure. Propagation dampens isolated matches and strengthens candidates embedded in a coherent cluster, improving robustness to feature noise and distractors.
(3) **Similarity refinement and smoothing.** Propagation effectively performs a smoothing operation over the graph, refining affinities so that items in the same underlying identity/instance manifold move closer in score. This reduces "ranking jitter" caused by local, unreliable distances.
(4) **Global consistency and transitive reasoning** Message passing enables transitive cues: if $q$ is close to $a$, and $a$ is strongly connected to $b$, then $b$ can be promoted even if $q$ and $b$ are not initially strong. This improves consistency across the retrieved set and helps recover hard positives.
(5) **Learned weighting of evidence.** Attention/gating in propagation can learn to prioritize reliable neighbors and downweight hubs or ambiguous nodes. This is an advantage over hand-crafted diffusion: the model can adapt propagation strength based on feature quality, edge confidence, or node density.

### 5.2 Research significance

In this paper, our contribution focuses on re-ranking, a post-processing stage that is orthogonal to feature learning. While [7, 20, 47, 54] aim to generate robust feature embeddings to improve pairwise distinctiveness, they do not explicitly exploit the underlying manifold structure of the data distribution during retrieval. In contrast, our method leverages this neighborhood structure to refine the ranking list regardless of the input features. Consequently, our approach is not a competitor to [7, 20, 47, 54] but a complement; it is model-agnostic and can be applied on top of the features generated by these methods to yield further performance gains.

## 6 CONCLUSION

In this paper, we revisit re-ranking methods and identify the primary challenge, *i.e.*, high computational complexity. To address this, we reformulate the re-ranking process using a Graph Neural Network (GNN). Specifically, we leverage a neighbor-based representation and employ a two-layer GNN to aggregate information from neighboring nodes. The inherent properties of GNNs enable efficient parallelization, making our approach hardware-friendly and suitable for acceleration. Extensive experiments on five benchmarks show that our method achieves competitive performance in both retrieval precision and running time compared to state-of-the-art approaches. We believe this work will contribute to the advancement of real-time retrieval tasks.

## ACKNOWLEDGMENTS

Z. Zheng acknowledges supports from Guangdong Basic and Applied Basic Research Foundation 2025A1515012281, Nanjing Municipal Science and Technology Bureau 202401035, University of Macau Advanced Research Institute in Hengqin, and the Macao Science and Technology Development Fund Grant FDCT/0043/2025/RIA1.

## REFERENCES

[1] Song Bai and Xiang Bai. 2016. Sparse Contextual Activation for Efficient Visual Re-Ranking. *IEEE Transactions on Image Processing* 25, 3 (2016), 1056–1069. doi:10.1109/TIP.2016.2514498.

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *ICLR*.

[3] De Cheng, Zhihui Li, Yihong Gong, and Dingwen Zhang. 2020. Fusion of multiple person re-id methods with model and data-aware abilities. *IEEE Transactions on Cybernetics* 50, 2 (2020), 561–571. doi:10.1109/TCYB.2018.2869739.

[4] Yoonki Cho, Woo Jae Kim, Seunghoon Hong, and Sung-Eui Yoon. 2022. Part-based pseudo label refinement for unsupervised person re-identification. In *CVPR*.

[5] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. 2007. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*.

[6] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. 2000. The mahalanobis distance. *Chemometrics and intelligent laboratory systems* (2000).

[7] Yongkang Ding, Jiechen Li, Hao Wang, Ziang Liu, and Anqi Wang. 2024. Re-ranking for image retrieval and transductive few-shot classification. *Complex and Intelligent Systems* (2024).

[8] Zhiyong Dou, Haotian Cui, Lin Zhang, and Bo Wang. 2020. Learning global and local consistent representations for unsupervised image retrieval via deep graph diffusion networks. *arXiv:2001.01284* (2020).

[9] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *ICML*.

[10] Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *IJCNN*.

[11] Chuchu Han, Zhedong Zheng, Kai Su, Dongdong Yu, Zehuan Yuan, Changxin Gao, Nong Sang, and Yi Yang. 2022. DMRNet++: Learning discriminative features with decoupled networks and enriched pairs for one-step person search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 6 (2022), 7319–7337.

[12] Alexander Hermans, Lucas Beyer, and Bastian Leibe. 2017. In defense of the triplet loss for person re-identification. *arXiv:1703.07737* (2017).

[13] Ahmet Iscen, Yannis Avrithis, Giorgos Tolias, Teddy Furon, and Ondřej Chum. 2018. Hybrid diffusion: Spectral-temporal graph filtering for manifold ranking. In *ACCV*.

[14] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. 2017. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *CVPR*.

[15] Deyi Ji, Haoran Wang, Hanzhe Hu, Weihao Gan, Wei Wu, and Junjie Yan. 2021. Context-Aware Graph Convolution Network for Target Re-identification. In *AAAI*.

[16] Ziqi Jin, Jinheng Xie, Bizhu Wu, and Linlin Shen. 2022. Weakly supervised pedestrian segmentation for person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 3 (2022), 1349–1362.

[17] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[18] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. 2022. Correlation verification for image retrieval. In *CVPR*.

[19] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2019. Deepgcns: Can gcns go as deep as cnns?. In *ICCV*.

[20] Haojie Li, Mingxuan Li, Qijie Peng, Shijie Wang, Hong Yu, and Zhihui Wang. 2023. Correlation-guided semantic consistency network for visible-infrared person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 6 (2023), 4503–4515.

[21] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*.

[22] Jinliang Lin, Zhedong Zheng, Zhun Zhong, Zhiming Luo, Shaozi Li, Yi Yang, and Nicu Sebe. 2022. Joint representation learning and keypoint detection for cross-view geo-localization. *IEEE Transactions on Image Processing* 31 (2022), 3780–3792.

[23] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. 2019. Improving person re-identification by attribute and identity learning. *Pattern Recognition* 95 (2019), 151–161.

[24] Chundi Liu, Guangwei Yu, Maksims Volkovs, Cheng Chang, Himanshu Rai, Junwei Ma, and Satya Krishna Gorti. 2019. Guided similarity separation for image retrieval. *NeurIPS* 32, 1556–1566.

[25] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. 2016. Large-scale vehicle re-identification in urban surveillance videos. In *ICME*.

[26] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. 2016. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *ECCV*.

[27] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021. Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In *WWW*. 3168–3177.

[28] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. 2016. Hierarchical gaussian descriptor for person re-identification. In *CVPR*.

[29] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. 2019. When does label smoothing help? *arXiv:1906.02629* (2019).

[30] Huy Nguyen, Kien Nguyen, Akila Pemasiri, Sridha Sridharan, and Clinton Fookes. 2025. Beyond geometry: The power of texture in interpretable 3D person ReID. *Computer Vision and Image Understanding* (2025), 104517.

[31] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. 2018. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*.

[32] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*.

[33] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*.

[34] Xuelin Qian, Yanwei Fu, Yu-Gang Jiang, Tao Xiang, and Xiangyang Xue. 2017. Multi-scale deep learning architectures for person re-identification. In *ICCV*.

[35] Danfeng Qin, Stephan Gammeter, Lukas Bossard, Till Quack, and Luc Van Gool. 2011. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*.

[36] Filip Radenović, Giorgos Tolias, and Ondřej Chum. 2019. Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 7 (2019), 1655–1668. doi:10.1109/TPAMI.2018.2846566.

[37] Xi Shen, Yang Xiao, Shell Xu Hu, Othman Sbai, and Mathieu Aubry. 2021. Re-ranking for image retrieval and transductive few-shot classification. *NeurIPS* (2021).

[38] Yantao Shen, Hongsheng Li, Shuai Yi, Dapeng Chen, and Xiaogang Wang. 2018. Person re-identification with deep similarity-guided graph neural network. In *ECCV*.

[39] Roee Shraga, Haggai Roitman, Guy Feigenblat, and Mustafa Canim. 2020. Ad hoc table retrieval using intrinsic and extrinsic similarities. In *WWW*. 2479–2485.

[40] Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*.

[41] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. 2021. Instance-level image retrieval using reranking transformers. In *ICCV*.

[42] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[43] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. 2020. Scan: Learning to classify images without labels. In *ECCV*.

[44] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. 2018. Learning discriminative features with multiple granularities for person re-identification. In *ACM MM*.

[45] Haochen Wang, Jiayi Shen, Yongtuo Liu, Yan Gao, and Efstratios Gavves. 2022. Nformer: Robust person re-identification with neighbor transformer. In *CVPR*.

[46] Luo Wang, Xueming Qian, Xingjun Zhang, and Xingsong Hou. 2019. Sketch-based image retrieval with multi-clustering re-ranking. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 12 (2019), 4929–4943.

[47] Shijie Wang, Jianlong Chang, Zhihui Wang, Haojie Li, Wanli Ouyang, and Qi Tian. 2024. Content-aware rectified activation for zero-shot fine-grained image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 6 (2024), 4366–4380.

[48] Tingyu Wang, Zhedong Zheng, Chenggang Yan, Jiyong Zhang, Yaoqi Sun, Bolun Zheng, and Yi Yang. 2021. Each part matters: Local patterns facilitate cross-view geo-localization. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 2 (2021), 867–879.

[49] Tingyu Wang, Zhedong Zheng, Zunjie Zhu, Yaoqi Sun, Chenggang Yan, and Yi Yang. 2024. Learning cross-view geo-localization embeddings via dynamic weighted decorrelation regularization. *IEEE Transactions on Geoscience and Remote Sensing* (2024).

[50] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. 2019. Ranked list loss for deep metric learning. In *CVPR*.

[51] Xiaodong Wang, Zhedong Zheng, Yang He, Fei Yan, Zhiqiang Zeng, and Yi Yang. 2023. Progressive local filter pruning for image retrieval acceleration. *IEEE Transactions on Multimedia* 25 (2023), 9597–9607.

[52] Xiaohan Wang, Linchao Zhu, Zhedong Zheng, Mingliang Xu, and Yi Yang. 2022. Align and tell: Boosting text-video retrieval with local alignment and fine-grained supervision. *IEEE Transactions on Multimedia* 25 (2022), 6079–6089.

[53] Yadi Wang, Hongyun Zhang, Duoqian Miao, and Witold Pedrycz. 2023. Multi-granularity re-ranking for visible-infrared person re-identification. *CAAI Transactions on Intelligence Technology* 8, 3 (2023), 770–779.

[54] Yu Wang, Yuying Zhao, Yi Zhang, and Tyler Derr. 2023. Collaboration-aware graph convolutional network for recommender systems. In *WWW*. 91–101.

[55] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2021), 4–24. doi:10.1109/TNNLS.2020.2978386.

[56] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xinbo Gao. 2017. Pairwise relationship guided deep hashing for cross-modal retrieval. In *AAAI*.

[57] Fan Yang, Ryota Hinami, Yusuke Matsui, Steven Ly, and Shin'ichi Satoh. 2019. Efficient image retrieval via decoupling diffusion into online and offline processing. In *AAAI*.

[58] Jinxi Yang, He Li, Bo Du, and Mang Ye. 2025. Cheb-GR: Rethinking K-nearest Neighbor Search in Re-ranking for Person Re-identification. In *CVPR*.

[59] Shuyu Yang, Yinan Zhou, Zhedong Zheng, Yaxiong Wang, Li Zhu, and Yujiao Wu. 2023. Towards unified text-based person retrieval: A large-scale multi-attribute and language search benchmark. In *Proceedings of the 31st ACM international conference on multimedia*. 4492–4501.

[60] Mingshan You, Jiao Yin, Hua Wang, Jinli Cao, Kate Wang, Yuan Miao, and Elisa Bertino. 2023. A knowledge graph empowered online learning framework for access control decision-making. *WWW* 26, 2 (2023), 827–848.

[61] Chao Yuan, Guiwei Zhang, Changxiao Ma, Tianyi Zhang, and Guanglin Niu. 2025. From poses to identity: Training-free person re-identification via feature centralization. In *CVPR*.

[62] Dingwen Zhang, Wenyuan Zeng, Jieru Yao, and Junwei Han. 2020. Weakly Supervised Object Detection Using Proposal- and Semantic-Level Relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–1. doi:10.1109/TPAMI.2020.3046647.

[63] Hao Zhang, Xin Chen, Heming Jing, Yingbin Zheng, Yuan Wu, and Cheng Jin. 2023. ETR: An Efficient Transformer for Re-Ranking in Visual Place Recognition. In *WCACV*.

[64] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. 2015. Scalable person re-identification: A benchmark. In *ICCV*.

[65] Zhedong Zheng, Tao Ruan, Yunchao Wei, Yi Yang, and Tao Mei. 2021. VehicleNet: Learning Robust Visual Representation for Vehicle Re-Identification. *IEEE Transactions on Multimedia* 23 (2021), 2683–2693. doi:10.1109/TMM.2020.3014488.

[66] Zhedong Zheng, Xiaohan Wang, Nenggan Zheng, and Yi Yang. 2022. Parameter-efficient person re-identification in the 3d space. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[67] Zhedong Zheng, Yunchao Wei, and Yi Yang. 2020. University-1652: A Multi-view Multi-source Benchmark for Drone-based Geo-localization. In *ACM MM*.

[68] Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, Mingliang Xu, and Yi-Dong Shen. 2020. Dual-path convolutional image-text embeddings with instance loss. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, 2 (2020), 1–23.

[69] Zhedong Zheng, Liang Zheng, and Yi Yang. 2017. A discriminatively learned cnn embedding for person reidentification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 1 (2017), 1–20. doi:10.1145/3159171.

[70] Zhedong Zheng, Liang Zheng, and Yi Yang. 2019. Pedestrian Alignment Network for Large-scale Person Re-Identification. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 10 (2019), 3037–3045. https://doi.org/10.1109/TCSVT.2018.2873599

[71] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. 2017. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*.

[72] Bin Zhu, Tongkun Xu, Bolun Zheng, Quan Zhang, Yaoqi Sun, Anan Liu, Zhendong Mao, and Chenggang Yan. 2021. Evolution of ICTs-empowered-identification: A general re-ranking method for person re-identification. *Pattern Recognition Letters* 150 (2021), 94–100.

Received March 2025; revised xx xxxx; accepted xx xxxx