Received: 5 October 2022 | Revised: 10 November 2022 | Accepted: 5 December 2022

CAAI Transactions on Intelligence Technology

DOI: 10.1049/cit2.12151

ORIGINAL RESEARCH

# OSAP-Loss: Efficient optimization of average precision via involving samples after positive ones towards remote sensing image retrieval

Xin Yuan[1,2] | Xin Xu[1,2] | Xiao Wang[1,2] | Kai Zhang[1,2] | Liang Liao[3] | Zheng Wang[4] | Chia-Wen Lin[5]

[1]School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China

[2]Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, China

[3]Nanyang Technological University, Singapore, Singapore

[4]School of Computer Science, Wuhan University, Wuhan, China

[5]Department of Electrical Engineering, Industrial Technology Research Institute, National Tsing Hua University, Hsinchu, China

**Correspondence**

Xin Xu, School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China.
Email: xuxin@wust.edu.cn

**Funding information**

National Natural Science Foundation of China, Grant/Award Numbers: U1803262, 62176191, 62171325; Nature Science Foundation of Hubei Province, Grant/Award Number: 2022CFB018; Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Grant/Award Number: ZNXX2022001

## Abstract

In existing remote sensing image retrieval (RSIR) datasets, the number of images among different classes varies dramatically, which leads to a severe class imbalance problem. Some studies propose to train the model with the ranking-based metric (e.g., average precision [AP]), because AP is robust to class imbalance. However, current AP-based methods overlook an important issue: only optimising samples ranking before each positive sample, which is limited by the definition of AP and is prone to local optimum. To achieve global optimisation of AP, a novel method, namely Optimising Samples after positive ones & AP loss (OSAP-Loss) is proposed in this study. Specifically, a novel superior ranking function is designed to make the AP loss differentiable while providing a tighter upper bound. Then, a novel loss called Optimising Samples after Positive ones (OSP) loss is proposed to involve all positive and negative samples ranking after each positive one and to provide a more flexible optimisation strategy for each sample. Finally, a graphics processing unit memory-free mechanism is developed to thoroughly address the non-decomposability of AP optimisation. Extensive experimental results on RSIR as well as conventional image retrieval datasets show the superiority and competitive performance of OSAP-Loss compared to the state-of-the-art.

**KEYWORDS**

computer vision, image retrieval, metric learning

## 1 | INTRODUCTION

In recent years, with the investment in scientific research of satellite remote sensing technology, a large amount of relevant remote sensing data has been produced. As a result, numerous application research studies have been derived, among which remote sensing image processing applications have attracted wide attention. At the same time, these applications are in urgent need of effective remote sensing image retrieval (RSIR) techniques, which refers to the task of ranking semantically matched or similar images in a large remote sensing image database based on their relevance to the query [2–4]. RSIR has

been studied for many years, and its core is how to quickly and accurately find the target image from a large number of remote sensing images. The performance of the predicted results in RSIR is often evaluated using ranking-based metrics, for example, the standard recall at $k$ (Recall@k) [5], the normalised discounted cumulative gain (NDGG) [6], average precision (AP) [7, 8], and spearman coefficient [9]. Nevertheless, it is challenging to design a high-performance method for RSIR due to the complex and variable geographic information contained in remote sensing images.

With the continuous research and development of deep learning, end-to-end training in RSIR has become the de facto choice. Although deep neural networks (DNNs) combined with deep metric learning have achieved some success for RSIR, they are tenuous to class imbalance [10] (i.e. the number of images among different classes varies dramatically in RSIR datasets), which will lead to performance degradation. As the ranking evaluation metrics are robust to class imbalance according to their ranking-based error definition [11]. It has been proposed to directly optimise the well-defined ranking-based evaluation metrics to deal with the class imbalance problem. This intuition appears attractive, but it is notoriously difficult due to two major challenges posed by these evaluation metrics: (i) non-differentiability and (ii) non-decomposability. To tackle these problems, intuitively surrogate loss functions such as 0–1 loss [12, 13], the area under the ROC curve [14, 15], and cross entropy [16] were utilised extensively. These loss functions are decomposable, that is, they can be decomposed over every training sample. However, these attempts were merely made by optimising a structured hinge-loss upper bound to such evaluation metrics [6, 8] rather than the metrics themselves. Recently, there have been many studies using asymptotic methods to directly optimise, for example, binning approaches [17–21], neural networks [22], and blackbox methods [23, 24]. These methods provide an elegant upper bound for such metrics, but they are generally coarse approximations. Consequently, some efforts have attempted to design more accurate and smooth approximations [17–22, 25]. All studies conducted over many years have greatly promoted the optimisation of ranking-based evaluation metrics for vision tasks, such as conventional image retrieval (CIR) [26–30], face recognition [31–33], person/vehicle re-identification [34–39], and object detection [11] etc. Among these evaluation metrics, the AP is a pivotal evaluation metric in RSIR.

Therefore, we consider AP optimisation as a solution to overcome the class imbalance problem in RSIR. However, there is an important issue with AP optimisation that has been overlooked by previous research: only optimising samples ranking before each positive sample, resulting in that all sample information after each positive sample is not fully exploited. This issue easily makes AP optimisation fall into a local optimum. Moreover, this issue has rarely received attention so far, and it is the first to focus on this issue in this paper. In addition, current AP-based methods adopt one special gradient strategy, which may assign a smaller gradient to positive or negative samples with higher scores referenced to a positive sample with a lower score. Therefore, it should have different gradients for different positive and negative samples, which facilitates adaptive optimisation and improves the robustness of the trained model. As shown in Figure 1, the intra-class diversity and inter-class similarity of remote sensing images are exceptionally significant, which poses a great challenge to the robustness of the learnt model. These issues and challenges affect the retrieval results of AP optimisation for RSIR.

In this paper, we tackle the above issues of directly optimising AP with the stochastic gradient descent (SGD) methods for RSIR, which not only address the non-differentiable and non-decomposable nature of ranking to AP, but also include the

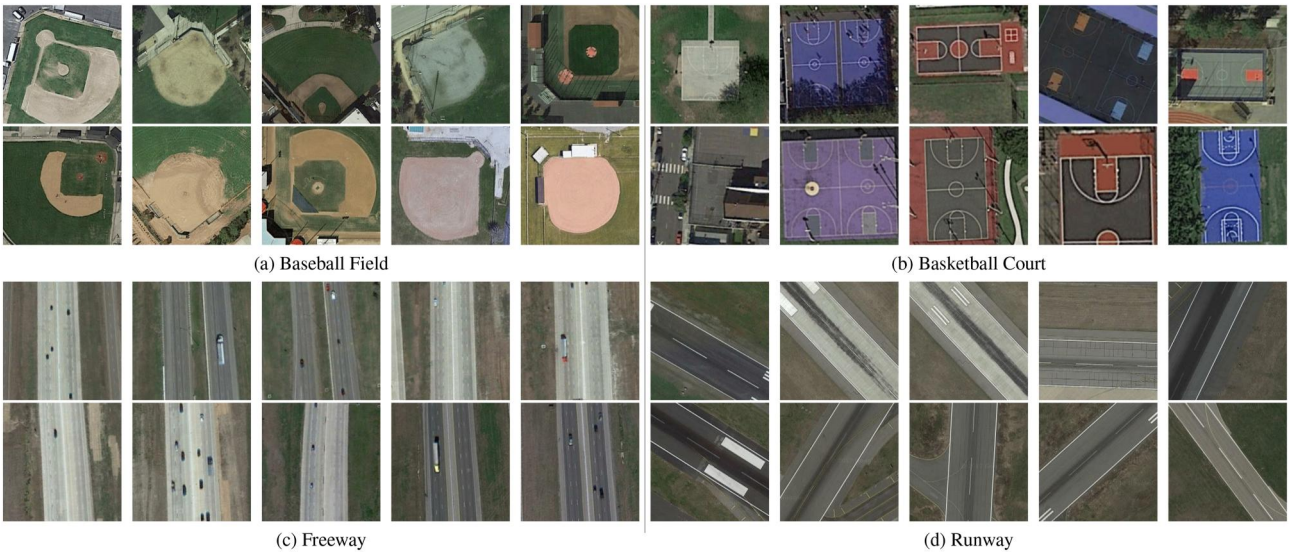



**FIGURE 1** Sample images of four different categories (e.g. baseball field, basketball court, freeway, and runway) from PatternNet [1] dataset. (a) Baseball field and (b) basketball court images, where the images with the same category show significant diversity. And (c) freeway and (d) runway images, where extreme similarity exists between the images of different categories.

specific optimisation limited by the AP definition. Existing AP-based methods have investigated and achieved some success in the non-differentiability and non-decomposability of AP, but improvement is still limited (detailed theoretical analysis is shown in Section 4). Therefore, a more accurate and effective optimisation of AP loss remains an open and explorable issue.

To this end, we propose a novel method, namely Optimising Samples after positive ones & AP loss (OSAP-Loss) for RSIR, which provides an efficient training objective for the optimisation and improvement of AP. Specifically, OSAP-Loss includes three components: $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$, $\mathcal{L}_{\mathrm{OSP}}$, and memory-free mechanism (MFM). First, to address the non-differentiability of AP and avoid the vanishing gradient of some samples, a novel superior ranking function (SRF) is proposed to enable AP loss differentiable while providing a tighter upper bound, thereby ensuring that it can be directly optimised using standard gradient descent methods, that is, $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$. Then, we introduce a novel loss, called Optimising Samples after Positive ones (OSP) loss (i.e. $\mathcal{L}_{\mathrm{OSP}}$), to reduce intra-class variability and inter-class similarity for further optimising $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$. More importantly, the $\mathcal{L}_{\mathrm{OSP}}$ considers the ranking of all positive and negative samples after each positive one and offers a more flexible optimisation approach for each sample, which maintains a margin between positive and negative samples. In this case, the initial version of OSAP-Loss (i.e. $\mathcal{L}_{\mathrm{OSAP}}$) is formulated, that is, by linearly combining $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ and $\mathcal{L}_{\mathrm{OSP}}$. To achieve a small deviation in AP optimisation, a graphics processing unit (GPU) MFM is developed to sidestep the hardware constraints of the GPU memory, which could thoroughly address the non-decomposability in AP optimisation. The MFM is inspired by the instance-level retrieval work [20], and we are the first to apply this technology to the field of RSIR. It is possible to use a very large batch size with several thousand high-resolution images on a single GPU. At last, $\mathcal{L}_{\mathrm{OSAP}}^{\mathrm{MFM}}$ is derived from $\mathcal{L}_{\mathrm{OSAP}}$ with MFM. This straightforward yet effective loss function can improve the robustness of the trained model for large-scale RSIR tasks. Meanwhile, we provide a theoretical analysis and discussion to verify the superiority and effectiveness of our proposed method.

In summary, the major contributions of this paper are as follows:

1) We propose shifting ranking orders rather than modifying the metric learning loss and minimising distances in the embedding space. In this paper, we propose a new method, named OSAP-Loss, which provides an efficient training objective for the optimisation and improvement of AP. To avoid the vanishing gradient of some samples, we design a novel SRF that is differentiable for the AP loss (i.e. $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$) while providing a tighter upper bound, thus ensuring that it can be directly optimised using standard gradient descent methods.
2) To reduce intra-class variability and inter-class similarity for further optimising $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$, we design a novel loss, for example, OSP loss, to involve all samples after positive ones while offering a more flexible optimisation approach for each sample for global ranking optimisation. In addition, we are the first to develop a GPU MFM to thoroughly address the non-decomposability of AP by sidestepping memory constraints of GPU in RSIR. The MEM enables the training of arbitrarily sized batches, which guarantees the global optimisation of AP and thereby improves the overall ranking performance.
3) A theoretical analysis is provided to show the superiority and effectiveness of our method. More importantly, we conduct extensive experiments on six image retrieval benchmarks, that is, three RSIR benchmarks containing UCMD [40], NWPU-RESISC45 [41], and PatternNet [1], as well as three CIR benchmarks, including CUB-200-2011 [42], Stanford Online Product (SOP) [43], and INaturalist-2018 [44]. Experimental results show that our proposed OSAP-Loss is on par with or superior to the state-of-the-art, which also demonstrates its effectiveness.

The rest of this paper is organised as follows. The relevant works on CIR, RSIR, deep metric learning and direct optimisation for the AP metric are discussed in Section 2. We introduce some preliminaries used in this work in Section 3 and describe the proposed method in Section 4. In Section 5, we elaborate experimental settings and extensive experimental results. Finally, conclusions are drawn in Section 6.

## 2 | RELATED WORK

In this section, we first review the related works of CIR in Section 2.1, and then we revisit the studies on RSIR in Section 2.2. Next, Section 2.3 provides the description and analysis of deep metric learning. Finally, we introduce the methods of direct optimisation for AP in deep metric learning in Section 2.4.

### 2.1 | Conventional image retrieval

CIR has been a fundamental and hot research topic in the field of information retrieval. It aims to find all images with relevant content to the query in the database. Existing efforts on CIR mainly focus on two categories: (1) the speed of retrieval and (2) the accuracy of retrieval. For the speed of retrieval, with the explosive growth of image content on the Internet, how to conduct fast and effective retrieval has emerged as major attention, where deep hashing [45–47] has become a leading technique for fast image retrieval. Previous studies mainly utilised hand-crafted image features like scale-invariant feature transform (SIFT) to learn hash functions for modelling data structures to preserve image similarities. Existing hashing methods directly extract features to learn hash codes or hash functions [48, 49] with the help of capable DNNs. For the accuracy of retrieval, it focuses on designing or learning more efficient image representations to achieve higher accuracy image retrieval performance. The earlier works aimed at obtaining compact image descriptors composed of multiple local features,

such as Fisher vectors [50] and vector of locally aggregated descriptors (VLAD) [51, 52]. In recent years, DNNs have made wonderful progress in CIR due to their powerful non-linear fitting and feature capturing capabilities. On the one hand, discriminative descriptors are generated by designing specific neural network structures. On the other hand, designing training objectives [43, 53–58] to achieve robust and generalised feature distributions is another highly typical research area, namely deep metric learning. It is worth noting that CIR can be extensively viewed as a problem of learning to rank. Meanwhile, the key to deep metric learning is to design a loss function that optimises a good ranking as opposed to classification, which is consistent with the objective of image retrieval. Therefore, image retrieval can be solved by combining deep metric learning with an appropriate ranking loss.

## 2.2 | Remote sensing image retrieval

RSIR is closely similar to CIR, except that the images retrieved are remote sensing images. In the last decade, RSIR has primarily focussed on the effective and discriminative feature extraction [2, 4, 59–62] and the dataset construction [1, 40, 41] of remote sensing image scenes. The field of feature extraction can be roughly divided into two categories: (1) hand-crafted features (e.g. SIFT [61], colour features [62], texture features [63], shape features [64], and spatial relationships [65] etc.); (2) deep features extracted by DNNs [2, 4, 59, 60]. Compared with hand-crafted features, it has become a consensus that deep learning can improve the accuracy of RSIR tasks, thanks to the tremendous feature capabilities of DNNs. Meanwhile, with the construction of research datasets in the field of remote sensing (e.g. RSIR datasets containing NWPU-RESISC45 [40], NWPUD [41], and PatternNet [1] etc.), which provides a large amount of training data for deep learning methods. Currently, the DNN-based RSIR method is to extract the features of remote sensing images with DNNs, and then train the model using classification loss [4, 66]. And deep metric learning methods have great potential in RSIR. The loss function plays a vital role in improving the discrimination and distribution of the learned features by DNNs. At present, some loss functions have been applied to RSIR with good results, including contrastive loss [67], triplet loss [68], N-Pair loss [69], Proxy NCA loss [70], lifted structured loss [43], and distribution structure learning loss (DSLL) [71]. However, these methods ignore the distribution of intra-class and the differences of inter-class. To this problem, Fan et al. [2] proposed distribution consistency loss (DCL) [2] to select multiple positive and negative samples from different classes to maintain intra-class compactness as well as inter-class differences by separating the negative samples from different classes by different distances. Despite these deep metric learning methods being novel and effective for RSIR, these loss functions belong to local optimisation and are challenging for high-resolution images in RSIR. As stated previously, image retrieval itself is a ranking problem, and the current loss function optimises merely the upper bound of the ranking loss. Therefore, a globally optimised loss function is necessary for RSIR in deep metric learning.

## 2.3 | Deep metric learning

Deep metric learning plays a critical role in many tasks, which has been studied for decades. It concentrates on constructing an effective feature space that effectively reflects the similarity or dissimilarity among images. The objective of deep metric learning is to minimise intra-class similarity while simultaneously maximising inter-class similarity. Metric learning has been extensively investigated by learning Mahalanobis distance functions [72] or projection matrix [73] before deep learning received widespread attention. Existing studies based on deep metric learning have mainly focussed on the designing of suitable loss functions to guide feature representation learning. For example, widely used pair-wise (e.g. contrastive loss [67]) and tuple-wise (e.g. triplet based loss [58, 68, 74–76] and n-tuple-based loss [69, 70, 77]) have been investigated for many years in the field of image retrieval. The essence of these losses is to construct the similarity structure between positive and negative samples by enlarging the inter-class distance and reducing the intra-class distance (e.g. Euclidean distance or Cosine distance). Nevertheless, these methods typically perform a local optimisation due to the fact that they act on a limited and fixed number of samples during the training phase. In addition, these methods need a series of iterative operations during training, including the repeatedly random sampling of challenging image pairs or tuples, especially time-consuming hard sample mining, as well as the calculation of losses and back-propagating gradients.

Moreover, a number of current studies [78–80] have pointed out that above-mentioned methods have some potential limitations resulting in the local optimisation. Firstly, it can be noticed that in most of these methods, only a proportion of informative samples is considered for constructing a similarity structure. As a result, numerous non-trivial samples are ignored, biasing the construction of the similarity structure. Secondly, intra-class data distribution properties are often ignored by most of these methods. Especially, these methods [43, 69, 75] attempt to pull the samples from the same class as closely as possible, and in extreme cases, compress the samples from the same class into a single point, which may easily degrade their similarity structure. Thirdly, these methods treat each sample with a specific gradient optimisation strategy, which forces the optimisation of the model to spend more capacity on some samples that have a minor effect on the similarity structure. Importantly, another often overlooked yet essential issue so far is that these loss functions only optimise the upper bound surrogate loss of true ranking loss [8, 23, 81–83]. Therefore, when using the ranking metric for evaluation, the minimum of the loss does not always correspond to the minimum of the true ranking loss (i.e. sub-optimal), namely the minimum of the loss does not guarantee that it corresponds to the minimum of the true ranking loss when using the ranking metric for evaluation. Furthermore, the common property of existing deep metric learning methods is that most of them are driven by minimising distance (maximising similarity) and thus overlook the importance of shifting ranking orders, which is crucial when evaluated with rank-based metrics. In this case, there have been extensive

research efforts [17, 20, 24, 83–85] focussing on the direct optimisation of ranking metric, such as AP.

## 2.4 | Direct optimisation for AP

AP is a widely used evaluation metric in many tasks, such as object detection, image retrieval, person/vehicle re-identification and etc. Consequently, AP is a good, direct, and effective choice as an optimisation goal. Recently, some methods [17, 20, 24, 83–85] have been developed to directly optimise AP as the training objective. Nevertheless, there are two major challenges for optimising the approximation of AP: non-differentiability and non-decomposability. To address these two challenges, there are many studies in image retrieval proposed and dedicated to AP optimisation.

For the first challenge, it has been actually studied for many years. On the one hand, one solution is to find the smooth surrogate losses for AP approximation. The widely used surrogate losses for image retrieval are often based on the familiar losses (e.g. contrastive loss [67], triplet loss [68], quadruplet loss [86], or n-tuple loss [69, 70, 77]) to enforce local ranking. These methods only provide a very coarse upper bound for optimising the AP, and require a complex sampling strategy and some experience tricks to be effective. To this end, studying and designing a smooth upper bound on AP has received many progressive investigations [8, 23, 81–83]. Initially, some works have adapted structured SVM models [8, 87] to reduce the complexity of the corresponding loss-enhanced inference or to adjust to weak supervision [81]. After that, there are many AP optimisation methods, for example, using a large LSTM [22] or blackbox optimisation [24] to approximate the ranking step. On the other hand, another option is to design smooth approximations of the ranking function. The most typical approach is the soft-binning techniques [17–21] via smoothing discrete approximation of similarity scores. In recent years, another approach has been explored directly to approximate the non-differentiable part of the ranking function, such as neural networks [22] or sigmoid functions [25, 84]. These approaches provide a more accurate approximation of AP by designing tight and smooth approximations of the ranking function. Although these two categories of methods provide applicable and smooth AP approximation, there are still some limitations. Specifically, the former methods provide subtle upper bounds, but it is mostly a coarse AP approximation; although the latter methods provide accurate AP approximation, they cannot guarantee to provide an upper bound for AP loss.

For the second challenge, the main reason lies in that the AP metric cannot be linearly decomposed into individual samples. Specifically, due to the limited available computational space, existing training methods are based on linearly dividing the whole samples into multiple batches, in which there will be a gap between the calculated AP of multiple batches and the AP of the whole samples. Consequently, it delivers an inconsistent AP gradient estimator. To address this challenge, effective batch sampling [5, 74, 88] or selecting informative samples [69, 88–90] makes the data distribution in a mini-batch as close as possible to the whole data distribution. And, Wang et al. [91] proposed the cross-batch memory method to store learned features and compute the global objective based on the assumption of the slow drift in learned features. Non-decomposability has been effectively handled in AP optimisation by increasing the batch size with a brute-force effort [17, 20, 23, 25]. Although this way is effective in mitigating non-decomposability and improving model performance, it introduces an important overhead in the computation and memory of GPU, including two main steps for the computation of AP loss and the update of back-propagation gradients. Therefore, Ramzi et al. [85] tried to enable good performance for AP optimisation with a simple but effective loss $\mathcal{L}_{\text{calibr}}$, which uses small batches without introducing any overhead. Although these methods have been proposed to alleviate the *decomposability gap* and achieve significant success, it does not completely and essentially solve this issue.

Although the above works have been proposed to tackle these two major challenges in AP optimisation, the improvement of non-differentiability and non-decomposability is still limited. In addition, there is an overlooked issue of AP optimisation: only optimising samples ranking before each positive sample, which results in that all sample information after each positive sample is not fully exploited. In other words, existing AP based losses do not achieve genuine global ranking optimisation. In order to overcome these limitations, this paper proposes a new optimised loss function, named OSAP-Loss, to achieve global ranking optimisation for RSIR. Furthermore, we also discuss and analyse the superiority and effectiveness of our proposed method theoretically.

# 3 | PRELIMINARIES

In this section, we introduce some fundamental preliminaries that will be used throughout the paper. We first offer the definition of the notations. Then, we review the calculation process of AP and the form of AP loss. Finally, we present the cautions of AP optimisation.

## 3.1 | Notations

Let $\Omega = \left\{\left(\mathbf{x}_j, y_j\right)\right\}_{j=1}^{N}$ be the retrieval set in the retrieval systems, where $\left(\mathbf{x}_j, y_j\right)$ indicates $j$th image and its corresponding relevant label $y_j \in \{1, 0\}$ related to the query. Note that there are a total of $M$ queries contained in $\Omega$, that is, $\mathcal{Q} = \left\{\mathbf{q}_i\right\}_{i=1}^{M} \subseteq \Omega$. For each query $\mathbf{q}_i$, the label $y_j$ is assigned a relevant value, that is, '1' if $\mathbf{x}_j$ is relevant to $\mathbf{q}_i$, and otherwise '0'. Thus, the retrieval set $\Omega$ can be spilt into the relevant $\mathcal{P}_i$ and irrelevant $\mathcal{N}_i$ sets, that is, $\Omega = \mathcal{P}_i \cup \mathcal{N}_i$, which are comprised by all the same class and different classes respectively. In detail, with respect to $\mathbf{q}_i$, $\mathcal{P}_i = \left\{\mathbf{x}_j \in \Omega | y_{j|\mathbf{q}_i} = 1\right\}$ and $\mathcal{N}_i = \left\{\mathbf{x}_j \in \Omega | y_{j|\mathbf{q}_i} = 0\right\}$. Note that each element $\mathbf{x}_j$ in $\Omega$ is mapped to a vectorial embedding $\mathbf{v}_j \in \mathbb{R}^d$, where $d$ is the

embedding size. In order to map all elements into the embedding space, we employ a DNN with weights $\theta$, that is, $f_\theta(\cdot)$. Hence, we use $\mathbf{V} = \left\{\mathbf{v}_j \in \mathbb{R}^d | \mathbf{v}_j = f_\theta(\mathbf{x}_j)\right\}_{j=1}^{N}$ to denote the vectorial embedding set.

## 3.2 | Average precision

AP is one of the most commonly used evaluation metrics for information retrieval tasks. It is a value of the area under the precision-recall curve. For a query $\mathbf{q}_i$, the set $S_\Omega$ of relevance scores of all elements in $\Omega$ are calculated by a selected similarity measure. In this paper, we adopt the cosine similarity, so $S_\Omega$ can be defined as follows:

$$\mathcal{S}_\Omega = \left\{ s(\mathbf{q}_i, \mathbf{x}_j) | s(\mathbf{q}_i, \mathbf{x}_j) = \left\langle \frac{\mathbf{v}_{q_i}}{\|\mathbf{v}_{q_i}\|} \cdot \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|} \right\rangle \right\}_{j=1}^{N} \tag{1}$$

where $\mathbf{v}_{q_i}$ and $\mathbf{v}_j$ are the vectorical embeddings of the query $\mathbf{q}_i$ and the element $\mathbf{x}_j$ in $\Omega$, $\mathcal{S}_\Omega = \mathcal{S}_{\mathcal{P}_i} \cup \mathcal{S}_{\mathcal{N}_i}$, and $\mathcal{S}_{\mathcal{P}_i} = \{s_k, \forall k \in \mathcal{P}_i\}$, $\mathcal{S}_{\mathcal{N}_i} = \{s_k, \forall k \in \mathcal{N}_i\}$ are the relevant and irrelevant score sets respectively. Each query $\mathbf{q}_i$ thus corresponds to the relevant score set $\mathcal{S}_{\mathcal{P}_i}$ and irrelevant score set $\mathcal{S}_{\mathcal{N}_i}$. Thus, we can get a ranking list of images

$$\mathrm{RList}^N(\mathbf{q}_i, f_\theta) = \left\{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_j, \cdots \mathbf{x}_N | \mathbf{x}_j \in \Omega\right\} \tag{2}$$

ordered by their similarities $\mathcal{S}_\Omega$ to query $\mathbf{q}_i$, where $N$ is the number of returned images and $f_\theta$ projects $\mathbf{x}_j$ to the feature space as vectorial embedding $\mathbf{v}_j$.

Next, some transformations are required to be formalised in ranking-based loss. Firstly, the difference transformation transfers the relevant score $s(\mathbf{q}_i, \mathbf{x}_j)$ to the difference form with respect to the query $\mathbf{q}_i$:

$$\forall j, k, \quad \delta_{j,k} = s(\mathbf{q}_i, \mathbf{x}_j) - s(\mathbf{q}_i, \mathbf{x}_k) \tag{3}$$

where $\mathbf{x}_j$ can be either an element of the relevant set $\mathcal{P}_i$ or of the irrelevant set $\mathcal{N}_i$. Note that $\mathbf{x}_k$ is always an element of the relevant set (e.g. $\mathbf{x}_k \in \mathcal{P}_i$), that is, $\mathbf{q}_i$ and $\mathbf{x}_k$ are of the same class. The relevant label transformation transfers the relevant labels $y_j$ to the corresponding pairwise relevant form:

$$\forall j, k, \quad \gamma_{j,k} = \mathbb{1} \tag{4}$$

where $\mathbb{1}$ is the indicator function, $\gamma_{j,k} = 1$ if $y_j = 1$, $y_k = 1$ and $\gamma_{j,k} = 0$ otherwise. Then, the ranking position of an element $\mathbf{x}_j$ before $\mathbf{x}_k$ in $\Omega$ can be defined as follows:

$$\begin{aligned} Rank(k, \mathcal{P}_i) &= 1 + \sum_{j \in \mathcal{P}_i, j \neq k} \mathbb{1}\{\delta_{j,k} > 0\} \\ Rank(k, \mathcal{N}_i) &= \sum_{j \in \mathcal{N}_i, j \neq k} \mathbb{1}\{\delta_{j,k} > 0\} \end{aligned} \tag{5}$$

where $Rank(k, \mathcal{P}_i)$ and $Rank(k, \mathcal{N}_i)$ represent the ranking positing of the element $\mathbf{x}_j$ in $\mathcal{P}_i$ and $\Omega$, and $Rank(k, \Omega) = Rank(k, \mathcal{P}_i) + Rank(k, \mathcal{N}_i)$. The AP of query $\mathbf{x}_i$ can be calculated as follows:

$$\begin{aligned} AP_i &= \frac{1}{|\mathcal{P}_i|} \sum_{k \in \mathcal{P}_i} precision(k) = \frac{1}{|\mathcal{P}_i|} \sum_{j \in \mathcal{P}_i} \frac{Rank(k, \mathcal{P}_i)}{Rank(k, \Omega)} \\ &= \frac{1}{|\mathcal{P}_i|} \sum_{k \in \mathcal{P}_i} \frac{Rank(k, \mathcal{P}_i)}{Rank(k, \mathcal{P}_i) + Rank(k, \mathcal{N}_i)} \\ &= \frac{1}{|\mathcal{P}_i|} \sum_{k \in \mathcal{P}_i} \frac{1 + \sum_{j \in \mathcal{P}_i, j \neq k} \mathbb{1}\{\delta_{j,k} > 0\}}{1 + \sum_{j \in \mathcal{P}_i, j \neq k} \mathbb{1}\{\delta_{j,k} > 0\} + \sum_{j \in \mathcal{N}_i, j \neq k} \mathbb{1}\{\delta_{j,k} > 0\}} \end{aligned} \tag{6}$$

## 3.3 | AP loss

Finally, we can formulate the AP loss $\mathcal{L}_{AP}$ by averaging all over queries, that is, the optimisation of the ranking problem can be formed as follows:

$$\min_\theta \mathcal{L}_{AP}(\theta) = 1 - AP(\mathbb{1}; \theta) = 1 - \frac{1}{M} \sum_{i=1}^{M} AP_i(\mathbb{1}; \theta) \tag{7}$$

where $\theta$ is the wights of the DNN and $\mathbb{1}$ corresponds to the indicator function. Note that $AP(\mathbb{1}; \theta)$ is non-differentiable with respect to $\theta$ due to the indicator function $\mathbb{1}$, which often uses the Heaviside step function:

$$\mathcal{H}(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases} \tag{8}$$

where its curve is almost all horizontal with zero or undefined gradient. Therefore, the specific smooth differentiable rank approximations need to be designed such that the standard gradient descent method can be used for optimisation.

### 3.4 | Cautions for AP optimisation

In order to achieve effective and better AP optimisation, some cautions for the AP itself are noteworthy. Below, we describe and analyse the key cautions for AP optimisation.

**Non-differentiability**. It is extremely difficult for AP optimisation due to the presence of the indicator function $\mathbb{1}$ in Equations (6) and (7). Specifically, It is the indicator function $\mathbb{1}$ that leads to the nature of non-convexity and non-differentiability in AP loss. Therefore, It is a critical challenge to design a solution that is differentiable, effective, and alternative to the indicator function $\mathbb{1}$ in AP optimisation. Meanwhile, it is currently still an open problem.

**Non-decomposability**. Using AP to evaluate the model performance must be in the whole test set. Therefore, AP essentially cannot linearly decompose into several batches. Limited by GPU memory resources, the current direct AP optimisation methods are based on mini-batch approximations. Consequently, this small min-batch approximation will result in a large deviation between the estimated mAP and the true mAP, that is, the above referred to as the *decomposability gap* (detailed presentation shown in Figure 2).

**Limited optimisation by AP definition**. AP is a metric that belongs to the ranking-based metrics, which is brittle due to the high probability of ties happening (shown in Figure 3). The main reason for this limitation lies in that the size of existing datasets is so large that there is a high degree of inter-class similarity. In addition, once the positive sample score is higher than all the negative sample scores, the AP score does not change regardless of the difference between the positive and negative sample scores. In that case, if the difference between positive and negative sample scores is very slight, this case is not further optimised, which would make it highly sensitive on the test set due to the existence of potential shifts.

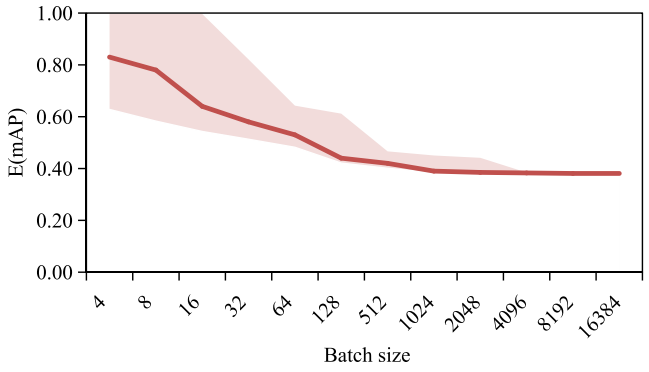



**FIGURE 2** Illustration of estimated mAP versus batch size, including the corresponding mean and standard deviation. Intuitively, it can be noticed that the smaller the batch size is, the larger the estimated mAP (i.e. corresponding AP optimisation) bias is. mAP, mean average precision.

Therefore, the avoidance of ties has attracted attention in some works [17, 18, 23]. Finally, AP only considers both positive and negative samples before each positive sample without involving the samples after it, which may lead to local optimisation due to insufficient utilisation of sample information. Therefore, it is necessary to make full use of all samples' information for global optimisation.

## 4 | METHODOLOGY

In this section, we introduce the optimisation solution for the previously mentioned problems of AP-based methods. Additionally, we provide an analysis of the loss functions associated with our proposed OSAP-Loss, such as SmoothAP [25] and ROADMAP [85]. The framework of our method is illustrated in Figure 4.

### 4.1 | Superior ranking function

In Equation (7), it contains a discontinuous ranking function (i.e. the indicator function $\mathbb{1}$) that is non-differentiable. To address this issue, we design a novel SRF, which provides a differentiable approximation and guarantees a tighter upper bound of the AP loss. The SRF ensures robust training directly optimised using standard gradient descent methods. Specifically, we use different operations for $Rank(k, \mathcal{P}_i)$ and $Rank(k, \mathcal{N}_i)$ in Equation (6), by defining two functions $\mathcal{H}_+(\cdot)$ and $\mathcal{H}_-(\cdot)$.

For $Rank(k, \mathcal{P}_i)$, similar to ref. [85], $\mathcal{H}_+(\cdot)$ remains to be the Heaviside step function, that is, $\mathcal{H}_+(z) = \mathcal{H}(z)$ (see Figure 5a). The purpose of this operation is to avoid contradictory gradient flow for some positive samples.

For $Rank(k, \mathcal{N}_i)$, we define a simple yet effective surrogate $\mathcal{H}_-(z)$ (see Figure 5b) as follows:

$$\mathcal{H}_-(z) = \begin{cases} 0.5 \cdot \left( \dfrac{\frac{z}{\gamma}}{1 + \left|\frac{z}{\gamma}\right|} + 1 \right) & z \le 0 \\ 0.5 \cdot \left( \dfrac{\frac{z}{\gamma}}{1 + \left|\frac{z}{\gamma}\right|} + 1 \right) + \delta & z > 0 \end{cases} \tag{9}$$

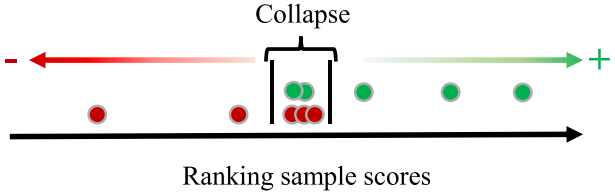



**FIGURE 3** Ranking sample scores may collapse when many ties happen due to the positive and negative samples with close scores. Green and red circles indicate positive and negative samples respectively.

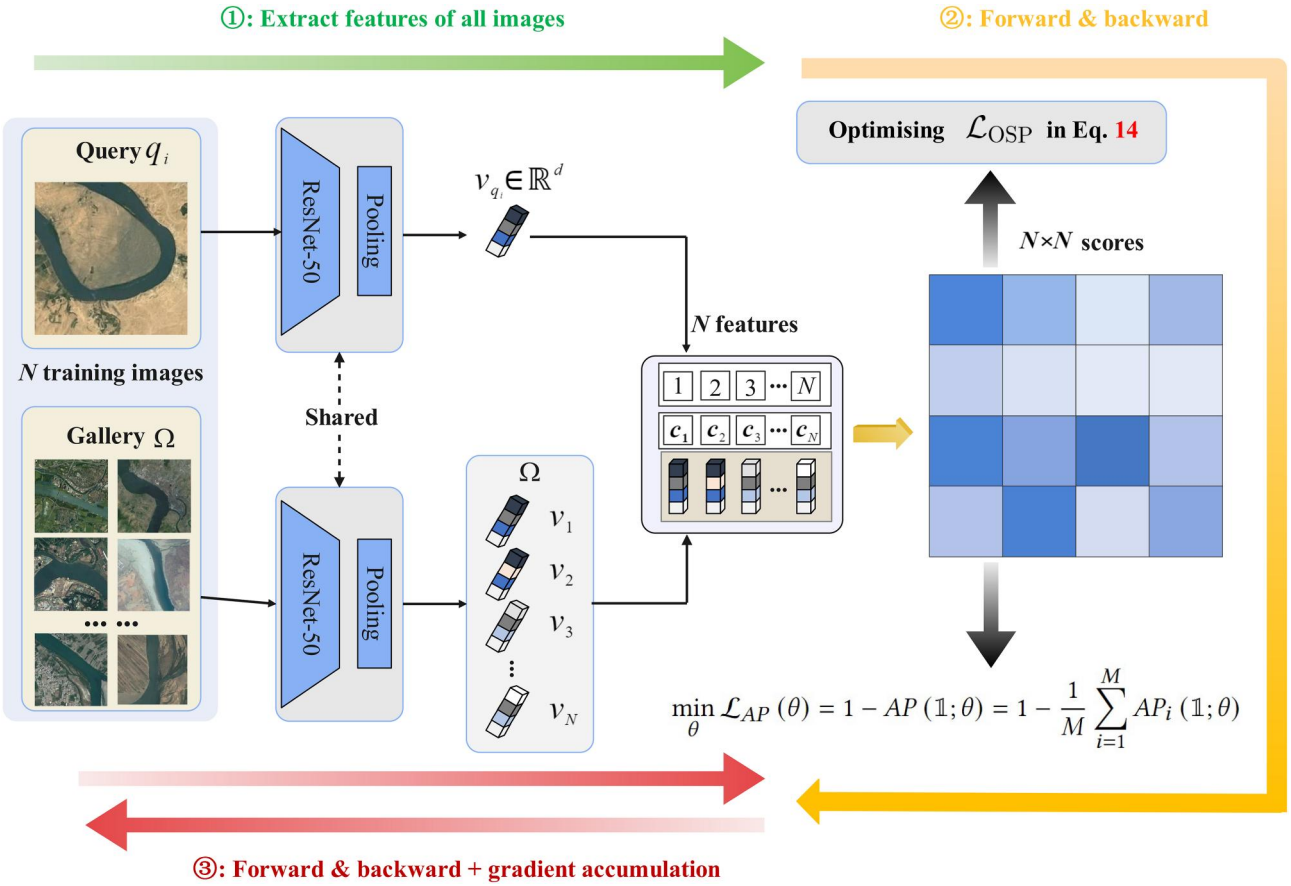
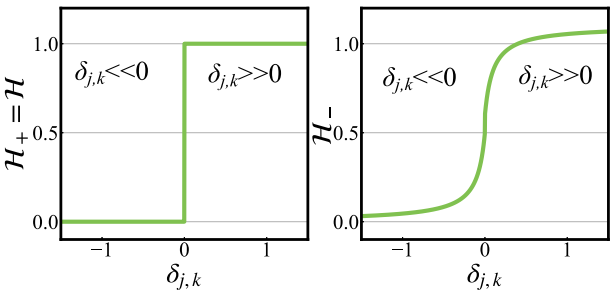



**FIGURE 4** The framework of our proposed OSAP-Loss employs the commonly used ResNet-50 [92] as the backbone. During the training process, $N$ training images include an image as the query $q_i$ and the rest images as the gallery set Ω. The optimisation procedure consists of three steps. For the **first step**, the network extracts the features of all images and then discards the intermediate tensors in the memory. For the **second step**, we first calculate the similarity matrix $\mathcal{S}_\Omega$ (Equation 1), calculate the OSAP-Loss $\mathcal{L}_{\text{OSAP}}$ containing two parts: $\min_\theta \mathcal{L}_{AP}(\theta)$ and $\mathcal{L}_{\text{OSP}}$, finally calculate the gradient of the loss regarding the features. For the **last step**, in order to continue backpropagation through the network, the network extracts the features from an image again, saving the intermediate tensors this time. Before the network weights are eventually updated, the gradients are accumulated, one image at a time. *Best viewed in colour.*




(a) $\mathcal{H}_+(z) = \mathcal{H}(z)$ in Eq. 8 (b) $\mathcal{H}_-(z)$ in Eq. 9

**FIGURE 5** Illustration of proposed superior ranking function for substituting the Heaviside step function: (a) $\mathcal{H}_+(z)$ and (b) $\mathcal{H}_-(z)$. In this case, $\mathcal{L}_{\text{AP}}^{\text{SRF}}$ becomes an upper bound of $\mathcal{L}_{\text{AP}}$, and $\mathcal{H}_-(z)$ ensures the correct gradient flow as well as avoiding gradient vanishing.

where $\gamma$ and $\delta$ are hyperparameters, and $\gamma$ is a scale factor and $\delta$ is an offset to make $\mathcal{L}_{AP}$ have an upper bound. It is worth noting that $\mathcal{H}_-(z)$ also avoids the vanishing gradient problem for negative samples after positive samples (see the left graph in Figure 6) According to Equation (9), can achieve a superior approximation $Rank(k,\mathcal{N}_i)_{SRF} = \sum_{j\in\mathcal{N}_i, j\neq k}\mathcal{H}_-(\delta_{j,k})$, obtaining the superior AP loss approximation as follows:

$$\mathcal{L}_{\text{AP}}^{\text{SRF}} = 1 - \frac{1}{M}\sum_{i=1}^{M}\frac{1}{|\mathcal{P}_i|}\sum_{k\in\mathcal{P}_i}\frac{Rank(k,\mathcal{P}_i)}{Rank(k,\mathcal{P}_i) + Rank(k,\mathcal{N}_i)_{SRF}} \tag{10}$$

## 4.2 | Memory-free mechanism

In Equation (7), $AP$ is linearly decomposable with respect to queries $\mathbf{q}_i$, while $AP_i$ is non-decomposable with respect to samples. For a query $\mathbf{q}_i$, the *decomposability gap* (i.e. $DG_{AP}$) is defined in ref. [85] as follows:

$$DG_{AP}(\theta) = \frac{1}{K}\sum_{b=1}^{K} AP_i^b(\mathbb{1};\theta) - AP_i(\theta) \tag{11}$$

where $K = N/M$, $N$ is the size of the retrieval set Ω and $M$ is the number of the queries $\mathcal{Q}$. To thoroughly address the

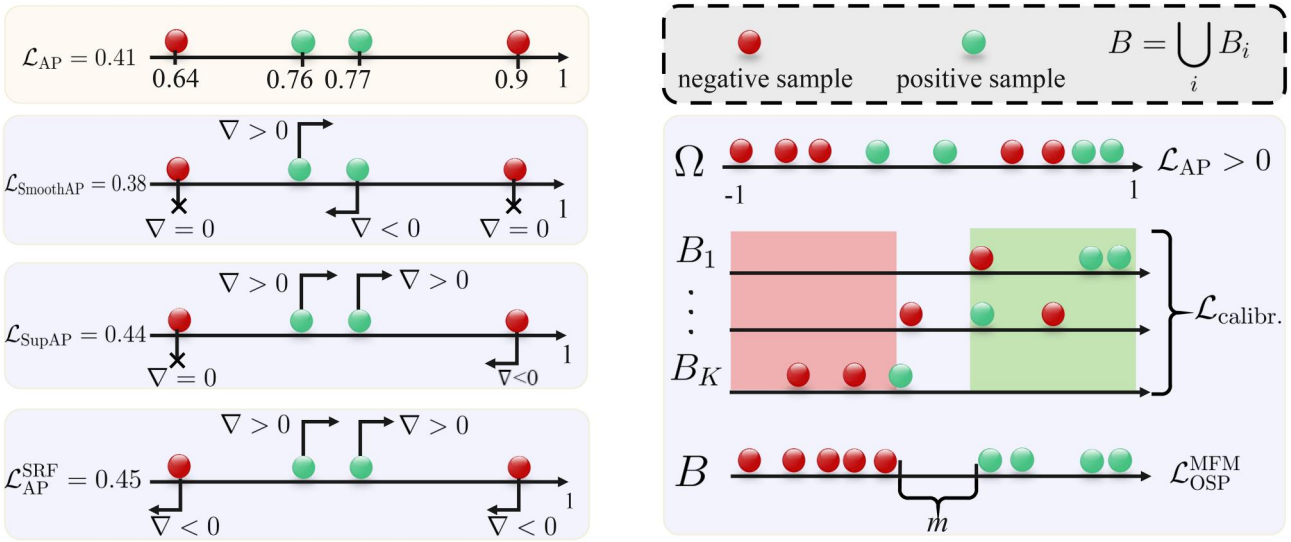



**FIGURE 6** Left: Comparison of gradient flow in AP loss for different samples in the left toy example. In contrast to SmoothAP [25] and SupAP [85], $\mathcal{L}_{OSAP} > \mathcal{L}_{AP}$ and correct gradient flow for all samples. Right: Comparison of approaches used to alleviate the decomposability gap in AP optimisation in the right toy example. Previous solutions are conducted by increasing the size of a batch with brute force, which has achieved some success. However, this method often requires a lot of computing costs. In each min-batch, $\mathcal{L}_{\text{calibr}}$ [85] solves this problem by dictating that the positive samples' scores be greater than $\alpha$ and the negative samples' scores are lower than $\beta$. On the contrary, Our $\mathcal{L}_{OSAP}^{\text{MFM}}$ employs a margin of $m$ to make it more closely resemble its positive set than its negative set. *Best viewed in colour*. AP, average precision.

*decomposability gap*, we introduce a GPU MFM by sidestepping the memory constraints of GPU following ref. [20]. The MFM allows the model to train with arbitrary resolution images and arbitrary batch size, which makes it possible to achieve optimal AP loss, and also means that $DG_{AP}$ can be thoroughly solved. The MFM is illustrated in Figure 4, and involves three main steps.

For the first step, the network extracts the features of all images and then discards the intermediate tensors in the memory. In the second step, we first calculate the similarity matrix $\mathcal{S}_\Omega$ (Equation 1), calculate the OSAP-Loss $\mathcal{L}_{\text{OSAP}}$ containing two parts: $\min_\theta \mathcal{L}_{AP}(\theta)$ and $\mathcal{L}_{\text{OSP}}$, finally calculate the gradient of the loss regarding the features, that is, $\frac{\mathcal{L}_{AP}}{\mathbf{v}_j}$. That is, in this step, we stop the network parameters updating by disrupting the back-propagation. Since a certain amount of feature vectors and similarity score matrices have to be saved, this step consumes a considerable amount of memory. During the last step, in order to continue back-propagation through the network, the network extracts the features from an image again, saving the intermediate tensors this time. Since this step consumes a lot of memory, we perform this operation on an image by image. Before the network weights are eventually updated, the gradients are accumulated in multiple steps, one image at a time.

Although the MFM sidesteps the constraint of the GPU memory resources to enable the training with a large batch size, it will contain an important overhead in memory. Especially, the last step of MFM is memory consuming, since the intermediate tensors after extracting the image features need to be stored. Therefore, this situation forces us to use the multistage back-propagation in the last step of MFM, that is, to forward one image at a time and then to back-propagate the corresponding gradient. This multistage back-propagation reduces the training speed of the network, but the utilisation of MFM decreases the number of the network iterations before the performance converge [20], and more importantly, brings a significant performance improvement (see Section 5.4). This paper focuses on performance optimisation. Since the total network training time does not increase much, the training time has not been explored accordingly in the experiments.

## 4.3 | Optimising samples after positive ones

Limited by AP definition, existing AP optimisation methods suffer from several problems leading to sub-optimisation (detailed discussion and analysis in Section 3.4). To address the brittleness around ties, we a score shift in order to keep a margin between the scores of different samples, which is similar to the triplet loss [68] and the circle loss [55]. Specifically, we apply a negative shift to the positive scores and a positive shift to the negative scores as follows:

$$\overleftrightarrow{s}(\mathbf{q}_i, \mathbf{x}_j) = \begin{cases} s(\mathbf{q}_i, \mathbf{x}_j) + \frac{\epsilon}{2} & \text{if } y_j = 0 \\ s(\mathbf{q}_i, \mathbf{x}_j) - \frac{\epsilon}{2} & \text{if } y_j = 1 \end{cases} \tag{12}$$

where $\epsilon$ is a tiny margin. Then, for the query $\mathbf{q}_i$, we use $\overleftrightarrow{\delta}_{j,k}$ instead of $\delta_{j,k}$ in Equation (3) to perform the difference transformation.

$$\forall j, k, \quad \overleftrightarrow{\delta}_{j,k} = \overleftrightarrow{s}(\mathbf{q}_i, \mathbf{x}_j) - \overleftrightarrow{s}(\mathbf{q}_i, \mathbf{x}_k) \tag{13}$$

Then, we consider all samples after each positive sample to fully exploit sample information for global optimisation. In addition, we also take into account that different samples should be assigned different gradients, thus allowing for high flexibility of optimisation. Therefore, we design a novel loss function to improve the optimisation of AP loss, that is, our proposed OSP loss.

$$\mathcal{L}_{\mathrm{OSP}} = \log\left(1 + \exp\frac{1}{|P|}\sum_{i}^{|P|}\sum_{j,j\neq i}^{|\Omega|} T\left\{y_{ij}\cdot w_{ij}^{p}\left[s_i - s_j\right]_{+} + \left(1 - y_{ij}\right)\cdot w_{ij}^{n}\left[\left(s_j + m\right) - s_i\right]_{+}\right\}\right) \tag{14}$$

$$\begin{aligned} w_{ij}^{p} &= \frac{\exp\left(\left[s_i - s_j\right]_{+}\right)}{\sum_{y_i=y_j=1}\exp\left(\left[s_i - s_j\right]_{+}\right)}, w_{ij}^{n} \\ &= \frac{\exp\left(\left[\left(s_j + m\right) - s_i\right]_{+}\right)}{\sum_{y_i=1,y_j=0}\exp\left(\left[\left(s_j + m\right) - s_i\right]_{+}\right)} \end{aligned} \tag{15}$$

where $y_{ij} = 1$ if $y_i = y_j = 1$ and $y_{ij} = 0$ otherwise, $m$ is a score margin. $T$ is the temperature parameter that controls the scale of the sum of the score differences of all samples with respect to the query. $[\cdot]_+$ represents the non-negative operation. $w_{ij}^{p}$ and $w_{ij}^{n}$ inherit the advantage of relative score optimisation between positive and negative samples, which does not introduce any hyperparameters. And $\mathcal{L}_{\mathrm{OSP}}$ has three hyperparameters, that is, the tiny margin $\epsilon$, the score margin $m$ between positive and negative samples, and the temperature parameter $T$.

**Algorithm 1 Training with OSAP-Loss**

**procedure** TRAIN-OSAP($\Omega, C, M, n$)
  $\Omega$ : training images
  $C$ : class labels
  $M$ : mini-batch size
  $n$ : number of images per class in mini-batch

  $\theta \leftarrow$ initialized ▷ use ImageNet pre-trained parameters
  **for** iteration $\in [1, \cdots,$ number-of-iterations$]$ **do**
    $loss \leftarrow 0$ ▷ set batch loss to zero
    $B \leftarrow$ BATH-SAMPLER($\Omega, C, M, n$)
    **for** $(\mathbf{q}_i, \mathbf{x}_j) \in B \times B$ **do**
      compute $s(\mathbf{q}_i, \mathbf{x}_j)$
    **end for**
    **for** $\mathbf{q}_i \in B$ **do** ▷ use each image in the batch as query
      $loss \leftarrow loss + \mathcal{L}_{OSAP}$ ▷ OSAP-Loss
    **end for**
    $\theta \leftarrow$ MINIMIZE($\frac{loss}{|B|}$) ▷ Adam update
  **end for**
**end procedure**

## 4.4 | OSAP-loss

Finally, our proposed OSAP-Loss (i.e. $\mathcal{L}_{\mathrm{OSAP}}$) is linearly combined with $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ and $\mathcal{L}_{\mathrm{OSP}}$ via a relative weight $\omega$.

$$\mathcal{L}_{\mathrm{OSAP}} = (1 - \omega)\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}} + \omega\mathcal{L}_{\mathrm{OSP}} \tag{16}$$

To learn a deep model for image retrieval, we implement the OSAP-Loss based on SGD and min-batch. The training procedure using OSAP-Loss is given in Algorithm 1. Furthermore, we have added the MFM to enable further improvement of $\mathcal{L}_{\mathrm{OSAP}}$, that is, $\mathcal{L}_{\mathrm{OSAP}}^{\mathrm{MFM}}$.

## 4.5 | Further discussion

Finally, we here compare with SmoothAP [25] and ROADMAP [85] ($\mathcal{L}_{\mathrm{SupAP}}$ and $\mathcal{L}_{\mathrm{calibr}}$ are two components of ROADMAP), which were recently proposed and are also most relevant to our proposed $\mathcal{L}_{\mathrm{OSAP}}^{\mathrm{MFM}}$. As shown in Figure 6, we have compared and analysed them accordingly, and there are three major differences: (1) it can well solve the issue of gradient vanishing; (2) it thoroughly solves the issue of non-decomposability in AP optimisation; (3) it solves the brittleness around ties in AP optimisation procedure, as well as enhances the separability of positive and negative samples.

Firstly, due to the sigmoid function, $\mathcal{L}_{\mathrm{SmoothAP}}$ will suffer from gradient vanishing and contradictory gradient flow for positive samples. Although the recently proposed $\mathcal{L}_{\mathrm{SupAP}}$ in ROADMAP addresses the contradictory gradient flow, it does not sufficiently tackle the gradient vanishing due to the use of the sigmoid function (see the left graph of Figure 6). Instead of continuing to optimise the sigmoid function, our designed SRF $\mathcal{H}_{-}(z)$ in $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ is optimised by using improved softsign function, which can eliminate the gradient vanishing more effectively while solving the contradictory gradient flow. Secondly, it has been demonstrated that large batch sizes will obtain better performance in refs. [17, 20, 23, 25, 85]. The reason lies in the fact that the AP is not linearly decomposed into multiple batches for optimisation. Therefore, the larger the batch size, the closer the estimated AP value is to the true AP, and the smaller the $DG_{AP}$ is. To overcome the memory constraints of GPU, our proposed MFM is to train the deep model with a large batch size, which can thoroughly solve the non-decomposability of AP (see the right graph of Figure 6).

Thirdly, for ranking-based metrics, one of the major drawbacks is the easy occurrence of ties, especially in large datasets. Our $\mathcal{L}_{\mathrm{OSP}}$ solves the brittleness around ties with score shift operation as well as improves the robustness of the learned model. These differences provide a significant performance improvement over several image retrieval datasets (Section 5).

Furthermore, we discuss the comparison of our method with other methods in terms of computational complexity. In total, we compare three types of losses, that is, proxy-based, pair-based and AP-based losses in Table 1. For proxy-based and pair-based losses, we can find that their computational complexity is much larger compared to AP-based losses. Meanwhile, in contrast to the general ranking function, they are all optimised based on local samples or sample pairs, which makes them easy to fall into local optimisation. For QS-Suitable [83] and SoftBin* [20], although they use AP as the optimisation objective and have a relatively small computational complexity, they do not employ a general ranking function, which leads to a very coarse AP approximation. Therefore, it is necessary to use an appropriate general ranking function for AP optimisation. With the help of the general ranking function, the AP-based losses can be continuously optimised towards the global optimum. However, these methods (BlackBox [24], PNP-D$_q$ [84], FastAP [17], and SoDeep [22]) perform a coarse AP approximation even by operating a general ranking function. Aiming to design tighter smooth AP approximation (SmoothAP [25] and SupAP [25]), our $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ proposed to design smooth differentiable approximation of the ranking function, where the core idea is similar to SupAP [85] with the same computational complexity $(O((\mathcal{N}+\mathcal{P})\mathcal{P}))$. We can see that the computational complexity of SmoothAP [25] is $O(\mathcal{M}^2)$, which is larger than that of SupAP [25] and our $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$. Compared to these methods, it is noted that our method is superior to both of them, as already discussed and analysed above.

## 5 | EXPERIMENTS

In this section, we first introduce the implementation details of our proposed method, the test protocol, and the benchmarks used for experiments, including three remote sensing image datasets and three CIR datasets. Then, we provide the relevant ablation study results and give a detailed analysis. Finally, we compare OSAP-Loss with several state-of-the-art methods.

### 5.1 | Datasets

We evaluate OSAP-Loss on the following three RSIR datasets (UCMD [40], NWPUD [41], and PatternNet [1]) and three

**TABLE 1** Comparison of the computational complexity for three types of losses, including proxy-based, pair-based and AP-based loss functions

| Loss | Type | Computational complexity | General ranking (AP approximation) |
|---|---|---|---|
| Proxy-Anchor [80] | Proxy | $O(\mathcal{MC})$ | ✗(–) |
| Proxy-NCA [70] | | $O(\mathcal{MC})$ | ✗(–) |
| SoftTriple [76] | | $O(\mathcal{MCU}^2)$ | ✗(–) |
| Contrastive [93, 94] | Pair | $O(\mathcal{M}^2)$ | ✗(–) |
| Triplet (smart) [74] | | $O(\mathcal{M}^2)$ | ✗(–) |
| Triplet (semi-hard) [68] | | $O(\mathcal{M}^3/\mathcal{B}^2)$ | ✗(–) |
| *N*-pair [69] | | $O(\mathcal{M}^3)$ | ✗(–) |
| Lifted structure [43] | | $O(\mathcal{M}^3)$ | ✗(–) |
| QS-suitable [83] | AP | $O(\mathcal{N}\log\mathcal{P})$ | ✗(Coarse) |
| SoftBin* [20] | | $O(\mathcal{NP})$ | ✗(Coarse) |
| BlackBox [24] | | $O(\mathcal{N}\log\mathcal{N})$ | ✓(Coarse) |
| PNP-D$_q$ [84] | | $O(\mathcal{NP})$ | ✓(Coarse) |
| FastAP [17] | | $O((\mathcal{N}+\mathcal{P})\mathcal{L})$ | ✓(Coarse) |
| SoDeep [22] | | $O((\mathcal{N}+\mathcal{P})\mathcal{H}^2)$ | ✓(Coarse) |
| SmoothAP [25] | | $O(\mathcal{M}^2)$ | ✓(Smooth) |
| SupAP [85] | | $O((\mathcal{N}+\mathcal{P})\mathcal{P})$ | ✓(Smooth) |
| OSAP $(\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}})$ | | $O((\mathcal{N}+\mathcal{P})\mathcal{P})$ | ✓(Smooth) |

*Note*: For proxy based and pair based losses, the numbers of training samples, classes, batch size in each epoch, and proxies of each class are represented by $\mathcal{M}$, $\mathcal{C}$, $\mathcal{B}$, and $\mathcal{U}$ respectively. For AP based losses, $\mathcal{P}$ and $\mathcal{N}$ ($\mathcal{P}+\mathcal{N}=\mathcal{M}$) denote the number of relevant (positive) and irrelevant (negative) samples respectively. The number of bins is denoted by $\mathcal{L}$ for FastAP. For SoDeep, $\mathcal{H}$ denotes the hidden state size ($\mathcal{H}\approx\mathcal{N}$) in LSTM. SmoothAP, SupAP, and our $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ are smooth AP approximations, which makes the AP calculation more accurate in practice.

Abbreviation: AP, average precision.

CIR datasets (CUB-200-2011 [42], SOP [43], and INaturalist-2018 [44]). The number of images, classes, and the average number of images per class are statistics in Table 2. We have counted the number of images (# Imges), classes (# Classes), and images/class (# Avg) in the training set and testing set for each dataset. Next, we describe the datasets in more detail.

1) RSIR Datasets
   - **UCMD** [40] is collected by the University of California Merced from the United States Geological Survey (USGS) for land use and cover. This dataset has 2100 high-resolution images with 21 different scene classes, which are 256 × 256 pixels and are 0.3 m in spatial resolution. We follow the standard protocol [95], which uses 50% images of each class as the training set and the rest as the test set. The sample images of each class are shown in Figure 7.
   - **NWPU-RESISC45 (NWPUD)** [41] is collected by the North-western Polytechnical University from Google Earth. This dataset contains 31,500 images of 45 classes. Each image is also 256 × 256 pixels and varies from 0.2 to 30 m in spatial resolution. All images cover more than 100 countries. Following the standard data splitting protocol [41], 80% images for training and the rest of 20% images for evaluation. Figure 8 illustrates samples of each class.
   - **PatternNet** [1] is a large-scale high-resolution dataset collected from Google Earth imagery or via Google Map API for RSIR. This dataset includes 30,400 images with 38 classes, and there are 800 images of 256 × 256 pixels

| Dataset | | # Images | # Classes | # Avg |
|---|---|---|---|---|
| Remote sensing image retrieval datasets | UCMD train [40] | 1100 | 21 | 52.4 |
| | UCMD test [40] | 1100 | 21 | 52.4 |
| | NWPUD train [41] | 25,200 | 45 | 560.0 |
| | NWPUD test [41] | 6300 | 45 | 140.0 |
| | PatternNet train [1] | 24,320 | 38 | 640.0 |
| | PatternNet test [1] | 6080 | 38 | 160.0 |
| Conventional image retrieval datasets | CUB train [42] | 5864 | 100 | 58.6 |
| | CUB test [42] | 5924 | 100 | 59.2 |
| | SOP train [43] | 59,551 | 11,318 | 5.3 |
| | SOP test [43] | 60,502 | 11,316 | 5.3 |
| | INaturalist train [44] | 325,846 | 5690 | 57.3 |
| | INaturalist test [44] | 136,093 | 2452 | 55.5 |

**TABLE 2** Dataset composition for training and evaluation

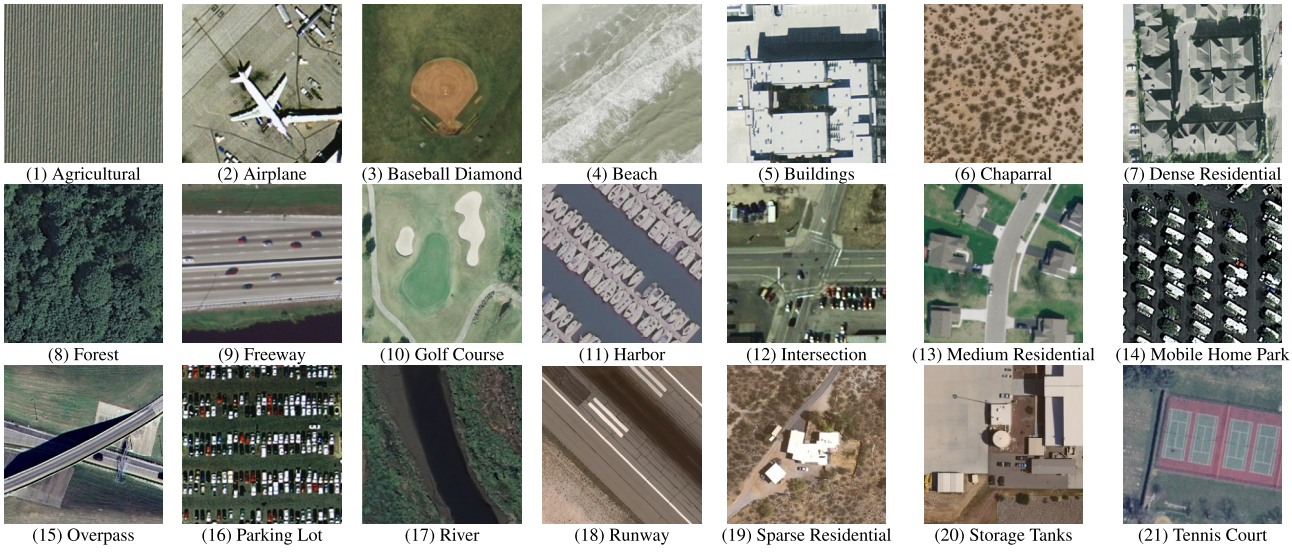



**FIGURE 7** Illustration of sample images of different scenes from UCMD dataset.

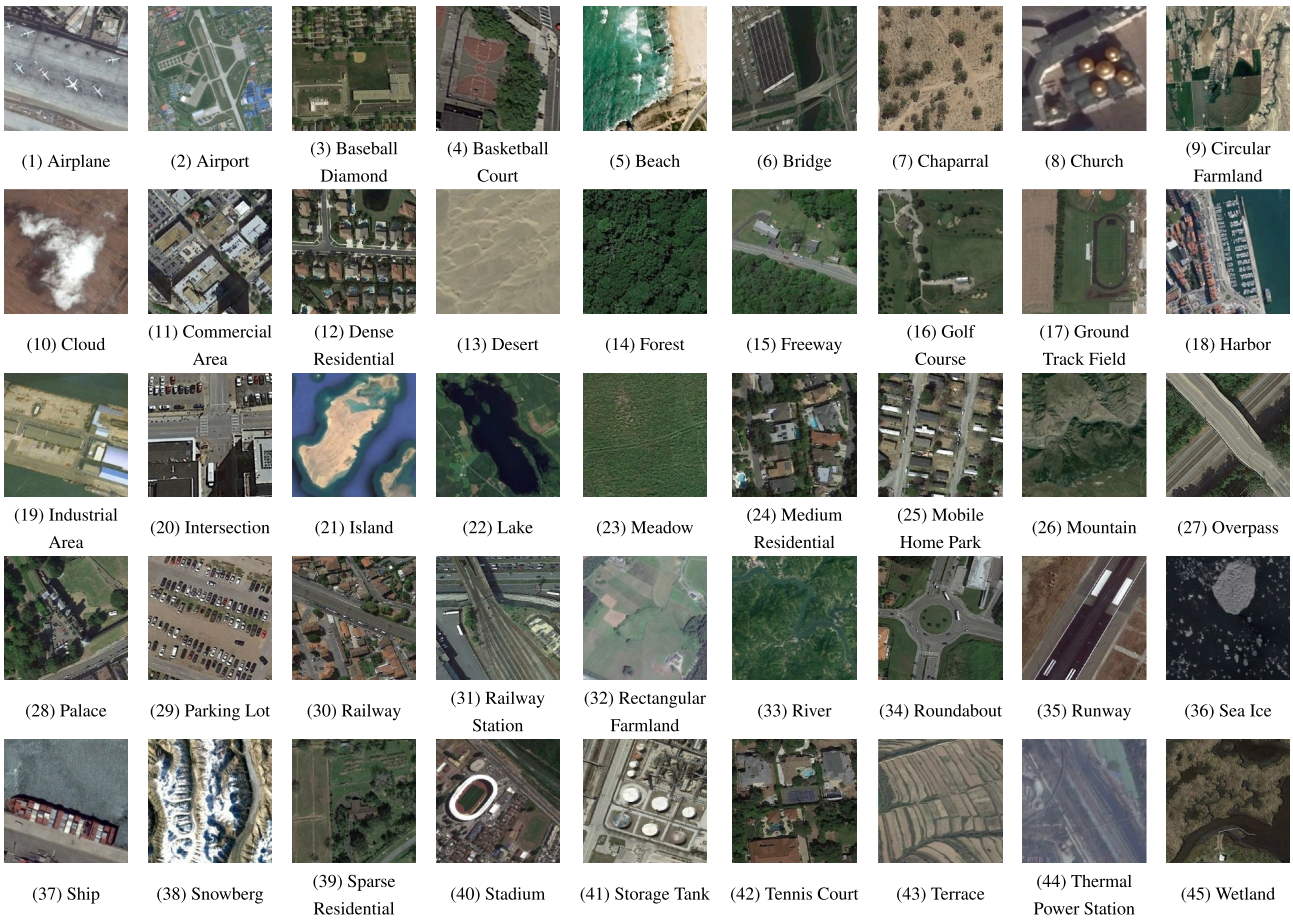



**FIGURE 8** Illustration of sample images of different scenes from NWPUD dataset.

per class. The spatial resolution ranges from 0.062 to 4.693 m. We follow the standard splitting protocol in ref. [1], that is, 80% training set and 20% testing set. Figure 9 shows corresponding sample images in the dataset.

2) CIR Datasets
   - **CUB-200-2011** [42] is a challenging dataset of 200 bird species. This dataset has 11,788 images. Following the standard protocol, we use the first 100 classes with 5864 images as the training set and the rest of the 100 classes with 5924 images as the testing set.
   - **Stanford Online Product (SOP)** is an online products dataset collected by Oh Song et al. [43] from eBay.com. This dataset contains 120,053 images with 22,634 classes. Referring to ref. [43], we use 11,318 classes with 59,551 images as the training set and 11,316 classes with 60,502 images as the testing set.
   - **INaturalist-2018** [44] is a real-world dataset with 461,939 images. This dataset includes 8142 iNaturalist species. Similar to ref. [25], we use 325,846 images of 5690 classes for training and 136,093 images of 2452 classes for testing.

## 5.2 | Implementation details

Our method includes training and evaluation, both implemented in the PyTorch framework. In our experiments, we use ResNet-50 [92] as the backbone. The model is initialized by the pre-trained parameters on ImageNet [96]. We use Adam [97] to optimize model training, and use standard data pre-processing procedure and augmentation strategy during training, that is, normalizing all images to 256 × 256 and randomly cropping them to 224 × 224 as input images. The crops are flipped horizontally with a 50% probability. During the evaluation, all images are normalized to 256 × 256 and then center cropped to 224 × 224. We use 333 as the fixed random seed in our all experiments to avoid seed performance fluctuations. We set the buffer size to 512 in MFM. For RSIR datasets, we train the models with a learning rate of $10^{-6}$ for 400 epochs. For CIR datasets, we also set the learning rate to $10^{-6}$ on CUB for 200 epochs. For SOP and INaturalist, we set the initial learning rate to $10^{-5}$ and drop the learning rate by 70% on 30 and 70 epochs. And the models are trained on SOP for 100 epochs and on INaturalist for 90 epochs. For all experiments in Section 5.4 and the models in Section 5.5, we use $\omega = 0.2$ for

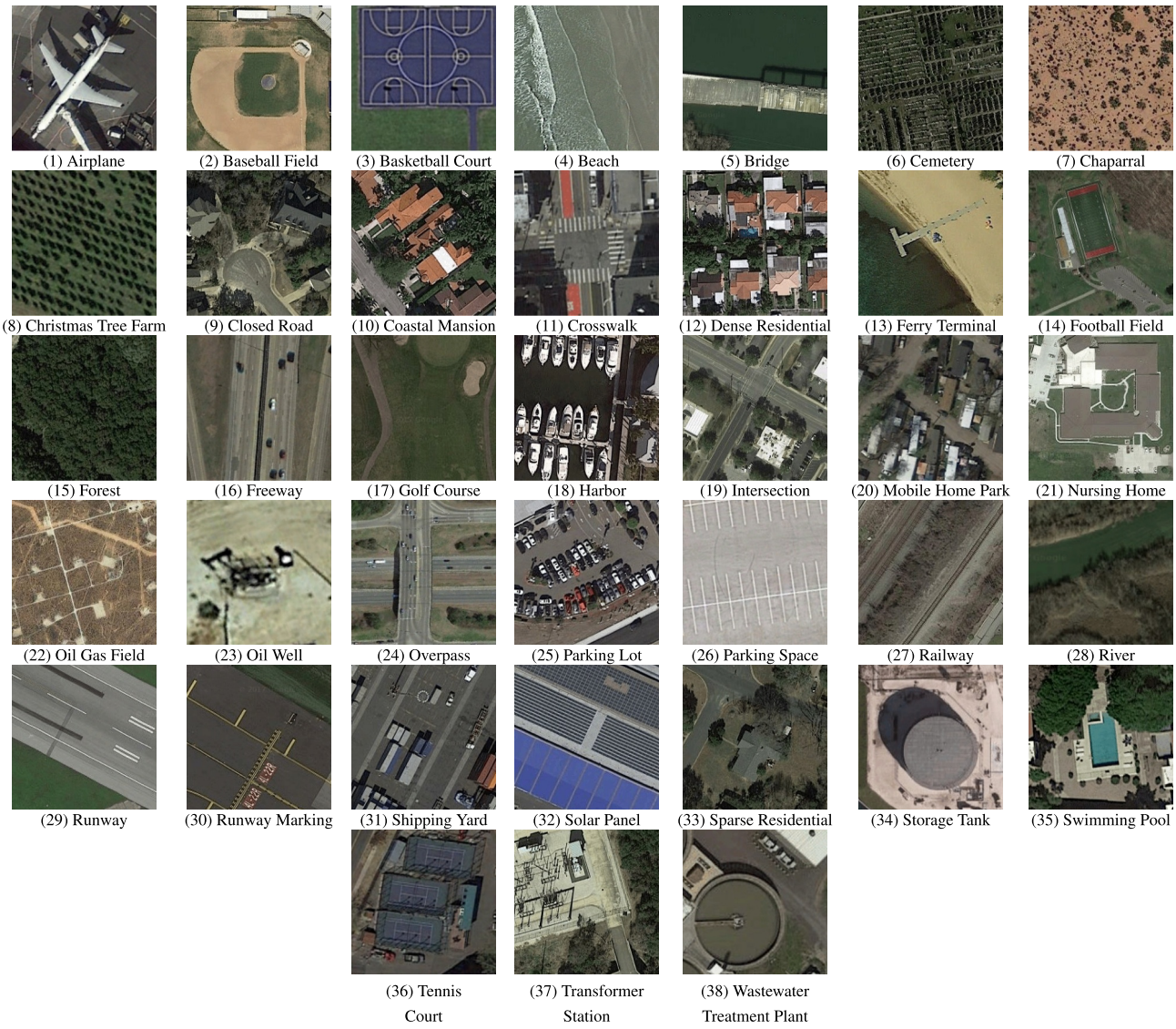



**FIGURE 9** Illustration of sample images of different scenes from PatternNet dataset.

$\mathcal{L}_{\text{OSAP}}$ in Equation (16), $\gamma = 0.001$ and $\delta = 0.65$ for $\mathcal{L}_{\text{AP}}^{\text{SRF}}$ in Equation (10), $\epsilon = 0.20$ on RSIR datasets, $\epsilon = 0.02$ on CIR datasets, and $m = 0.25$ for $\mathcal{L}_{\text{OSP}}$ in Equation (14).

## 5.3 | Test protocol

To evaluate the model performance, we describe here the used protocols in our experiments. For all image retrieval datasets, each sample from each class is utilised as the query $q_i$ in turn, and the retrieval set $\Omega$ is made up of all the remaining samples. For RSIR, we use standard precision at $k$ (P@k, precision of the top-k retrieval results) and mean average precision (mAP, detailed calculation given in Section 3). For CIR, we use standard recall at $k$ (R@k, hit rate of the top-k retrieval results) and mAP at $R$ (mAP@R) [79]. Below, we introduce the computation of the mentioned evaluation metrics.

**Precision@K** The Precision@K in Equation (17) is often used in the retrieval task. For each query, the Precision@K is precision at $k$, that is, precision of the top-$k$ retrieval results. The Precision@K is averaged over all the queries.

$$P@K = \frac{\sum_{i=1}^{M} P_k(i)}{M}, \text{ where } P_k(i) \text{ is the precision at } k. \quad (17)$$

**Recall@K** The Recall@K in Equation (18) is another evaluation metric for the retrieval task. For each query, the Recall@K is Recall at $k$, Recall@K = 1 if a positive sample appears in the top-$k$ retrieval results and Recall@K = 0 otherwise. The Recall@K is averaged over all the queries in Equation (18), where $R(i) = 1$ if a positive sample has a higher ranking than K, and $R(i) = 0$ otherwise.

$$R@K = \frac{\sum_{i=1}^{M} R(i)}{M} \tag{18}$$

**mAP@R** The mAP@R in Equation (19) is proposed in ref. [79]. It is less noisy and easily obtains the better performance of the model. The mAP@R is essentially a partial AP, which is an AP of $R$ retrieved positive samples with respect to a query.

$$mAP@R = \frac{\sum_{j=1}^{R} P(j)}{R}, \text{where } P(j) \text{ is the precision at } j. \tag{19}$$

## 5.4 | Ablation study

For a fair comparison, we conduct all our experiments here with the same settings (Backbone: ResNet-50; Batch size: 64; Embedding size: 512). Such comparisons allow us to directly observe the impact of different parts or parameters on the final performance.

### 5.4.1 | Impact of our designed components

To investigate more deeply the impact of our designed components, the relevant results of the ablation experiments are shown in Tables 3 and 4. Here, Table 3 shows the results on RSIR datasets and Table 4 shows the results on CIR datasets. In Table 3, it is clear to see that the performance of the SmoothAP baseline [25] is already impressive on three RSIR datasets. Meanwhile, we find that the performance of using $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ or $\mathcal{L}_{\mathrm{OSP}}$ alone does not perform better or slightly degrades on UCMD, NWPUD, and SOP. We presume that such performance degradation may be caused by local optimisation arising from the low diversity caused by the small number of samples within the batch. The performance improvement from using MFM alone and from using MFM on $\mathcal{L}_{\mathrm{OSAP}}$ shows that our presumption is plausible. In Table 4, it can be obviously observed that compared to SmoothAP [25], our method performs a great improvement by replacing the sigmoid with $\mathcal{H}_+$ and $\mathcal{H}_-$ for $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ in Equation (10) especially on CUB. Note that the use of $\mathcal{L}_{\mathrm{OSP}}$ in Equation (14) on $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ further boosts the experimental performance. After using MFM, we can see that $\mathcal{L}_{\mathrm{OSAP}}$ receives a further performance improvement of mAP/mAP@R on RSIR datasets (~3pt on UCMD, ~7pt on NWPUD, and 0.01pt on PatternNet) and CIR datasets (~6pt

**TABLE 3** Ablation study on our designed OSAP-Loss

| Dataset | SRF | OSP | MFM | mAP | P@5 | P@10 | P@50 | P@100 | P@1000 |
|---|---|---|---|---|---|---|---|---|---|
| UCMD | ✗ | ✗ | ✗ | 96.49 | 99.98 | 99.98 | 99.91 | 78.79 | 7.90 |
| | ✓ | ✗ | ✗ | 96.33 | 99.88 | 99.85 | 99.77 | 78.84 | 7.90 |
| | ✗ | ✓ | ✗ | 93.20 | 98.26 | 97.99 | 96.88 | 76.98 | 7.90 |
| | ✗ | ✗ | ✓ | 96.60 | 99.99 | 99.99 | 99.98 | **78.99** | 7.90 |
| | ✓ | ✓ | ✗ | 96.29 | 99.80 | 99.76 | 99.68 | 78.85 | 7.90 |
| | ✓ | ✓ | ✓ | **99.54** | **100.00** | **100.00** | **99.92** | 78.98 | **7.92** |
| NWPUD | ✗ | ✗ | ✗ | 92.53 | 95.35 | 95.16 | 94.37 | 93.00 | 13.89 |
| | ✓ | ✗ | ✗ | 90.38 | 94.31 | 93.99 | 92.94 | 91.15 | 13.88 |
| | ✗ | ✓ | ✗ | 88.94 | 92.84 | 92.40 | 90.95 | 89.87 | 13.90 |
| | ✗ | ✗ | ✓ | 97.79 | 98.49 | 98.50 | 98.30 | 97.87 | 13.89 |
| | ✓ | ✓ | ✗ | 92.60 | 94.83 | 94.48 | 93.72 | 92.63 | 13.90 |
| | ✓ | ✓ | ✓ | **99.73** | **100.00** | **100.00** | **99.91** | **99.80** | **13.93** |
| PatternNet | ✗ | ✗ | ✗ | 99.93 | 99.95 | 99.95 | 99.95 | 99.95 | 55.90 |
| | ✓ | ✗ | ✗ | 99.87 | 99.91 | 99.90 | 99.88 | 99.87 | 55.88 |
| | ✗ | ✓ | ✗ | 99.97 | 99.97 | 99.97 | 99.96 | 99.96 | 55.90 |
| | ✗ | ✗ | ✓ | 99.98 | 99.98 | 99.98 | 99.98 | 99.98 | 55.90 |
| | ✓ | ✓ | ✗ | 99.98 | 99.98 | 99.98 | 99.98 | 99.98 | 55.90 |
| | ✓ | ✓ | ✓ | **99.99** | **100.00** | **100.00** | **99.99** | **99.99** | **56.14** |

*Note*: The results are shown with the different designed components, and the SmoothAP baseline (i.e. SRF (✗), OSP (✗), and MFM (✗)) on UCMD, NWPUD, and PatternNet datasets. The results in bold indicate the best performance.

Abbreviations: mAP, mean average precision; MFM, memory-free mechanism; OSAP, Optimising samples after positive ones & average precision loss; OSP, Optimising Samples after Positive ones; SRF, superior ranking function.

**TABLE 4** Ablation study on our designed OSAP-Loss

| Method | SRF | OSP | MFM | SOP | | CUB | | INaturalist | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | R@1 | mAP@R | R@1 | mAP@R | R@1 | mAP@R |
| SmoothAP [25] | ✗ | ✗ | ✗ | 80.1 | 54.6 | 62.1 | 23.9 | 59.7 | 20.7 |
| OSAP | ✓ | ✗ | ✗ | 79.5 | 52.7 | 65.0 | 25.5 | 61.6 | 21.7 |
| | ✗ | ✓ | ✗ | 77.3 | 49.6 | 65.0 | 23.7 | 61.2 | 21.1 |
| | ✗ | ✗ | ✓ | 82.2 | 56.3 | 66.2 | 26.1 | 66.8 | 26.3 |
| | ✓ | ✓ | ✗ | 78.7 | 51.4 | 67.1 | **26.5** | 66.3 | 25.4 |
| | ✓ | ✓ | ✓ | **84.2** | **57.1** | **69.6** | 26.1 | **70.6** | **26.7** |

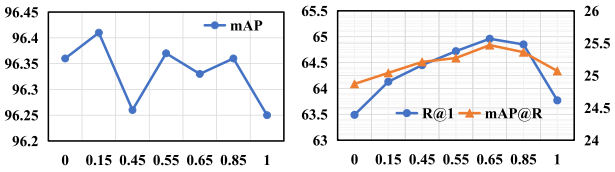
*Note*: The results are shown with the different designed components, and the SmoothAP baseline on SOP, CUB and INaturalist datasets. The results in bold indicate the best performance.

Abbreviations: mAP, mean average precision; MFM, memory-free mechanism; OSAP, Optimising samples after positive ones & average precision loss; OSP, Optimising Samples after Positive ones; SOP, Stanford Online Product; SRF, superior ranking function.

**TABLE 5** Ablation study over different hyperparameters of scale factor $\gamma$

| $\gamma$ | UCMD | CUB | |
|---|---|---|---|
| | mAP | R@1 | mAP@R |
| 1 | 93.24 | 40.19 | 11.43 |
| 0.1 | **96.62** | 52.58 | 18.58 |
| 0.01 | **96.62** | 61.21 | 23.87 |
| 0.001 | 96.33 | **64.30** | **24.38** |
| 0.0001 | 94.07 | 52.53 | 13.49 |

*Note*: The results are reported on UCMD and CUB. The results in bold indicate the best performance.




(a) offset $\delta$ on UCMD (b) offset $\delta$ on CUB

**FIGURE 10** Illustration of the effect of offset $\delta$ on (a) UCMD and (b) CUB datasets respectively.

on SOP, 0.6pt on CUB, and 1.3pt on INaturalist). In Table 3, $\mathcal{L}_{\text{AP}}^{\text{SRF}}$ has a greater contribution compared to $\mathcal{L}_{\text{OSP}}$, and the result can be further improved by adding $\mathcal{L}_{\text{OSP}}$ to $\mathcal{L}_{\text{AP}}^{\text{SRF}}$ on UCMD and NWPUD. Differently, $\mathcal{L}_{\text{AP}}^{\text{SRF}}$ and $\mathcal{L}_{\text{OSP}}$ contribute approximately the same to the performance improvement on CUB and INaturalist in Table 4. However, $\mathcal{L}_{\text{AP}}^{\text{SRF}}$ and $\mathcal{L}_{\text{OSP}}$ perform not especially well on SOP. We postulate that the reason may be # Avg is too small to optimise on SOP, as shown in Table 2. Note that the performance improvement of adding MFM is much more significant. This is explained by the fact that the MFM reduces the decomposability gap in Equation (11). We can see that MFM is very effective for improving performance and brings large gains both on RSIR datasets and CIR datasets.

### 5.4.2 | Impact of the hyperparameters

In this section, we will exploit the impact of the hyperparameters in $\mathcal{L}_{\text{OSAP}}$ without MFM. We analyse the impact of five hyperparameters, that is, the scale factor $\gamma$ and the offset $\delta$ in Equation (9), the tiny margin $\epsilon$ in Equation (12), the score margin $m$ and the temperature parameter $T$ in Equation (14), and the relative weight $\omega$ in Equation (16) on RSIR and CIR tasks respectively. Note that we only change one parameter at a time.

1) **The hyperparameters in SRF**. Here, we first exploit the effect of different hyperparameters in $\mathcal{L}_{\text{AP}}^{\text{SRF}}$, including the scale factor $\gamma$, and the offset $\delta$. We vary the two hyperparameters in a series of values to observe the effect on the experimental results, and the results are shown in Table 5 and Figure 10.
   - **Effect of scale factor $\gamma$**. $\gamma$ governs the smoothing of softsign that is used to approximate the indicator function in $\mathcal{L}_{\text{AP}}^{\text{SRF}}$. The scale factor is essential in lots of AP approximation methods [25, 84, 85]. It determines the region range size for gradient backpropagation of negative samples with high scores. We set $\gamma$ to different values and the performances are shown in Table 5. The ablation results show that a value of 0.001 leads to the best performance (i.e. mAP@R and R@1), which achieves the trade-off of AP approximation and gradient optimisation. Furthermore, the setting of the suitable scale factor is supposed to ensure a smooth AP approximation while also providing a large enough optimisation region for the gradient optimisation, thus achieving considerable optimisation.
   - **Effect of offset $\delta$**. $\delta$ controls the upper bound of AP in $\mathcal{H}_-$ and determines the degree of AP optimisation. We vary $\delta$ from 0 to 1 (with 0.05 as the interval) and visualise the results on UCMD and CUB in Figure 10. For UCMD, it can be observed that the change of $\delta'$ value has very little impact on the performance. For CUB, we can see that the performance is gradually improving when the value of $\delta$ from 0 to 0.65. Note that the best

performance is achieved when the value of $\delta$ is 0.65. There is a significant degradation when $\delta > 0.65$ possibly due to AP's overestimation.

2) **The hyperparameters in OSP**. Then, we exploit the effect of different hyperparameters in $\mathcal{L}_{\text{OSP}}$. It includes three hyperparameters, that is, the tiny margin $\epsilon$, the score margin $m$, and the temperature parameter $T$. $\epsilon$ is set to 0.20 on UCMD and 0.02 on CUB empirically in all experiments. Therefore, we study the effect of $T$ and $m$. And the results are shown in Table 6.
   - **Effect of score margin $m$**. $m$ determines the score margin between positive and negative samples. Here, we set $m$ to 0, 0.15, 0.25, 0.35, and 0.45. And we report the results in Table 6. For UCMD, the performance is relatively stable, holding at around 95.00%. For CUB, it can be observed that mAP@R is consistently stable at 23.7% in [0.25, 0.45], but R@1 increases and then decreases in [0.25, 0.45]. And R@1 and mAP@R achieve the best results (65.0% R@1 and 23.7% mAP@R) when $m$ is 0.25.
   - **Effect of temperature parameter $T$**. $T$ determines the scale of the sum of optimisation scores of positive and negative samples. It is critical in optimisation of $\mathcal{L}_{\text{OSP}}$. We vary $T$ from 45 to 85 (with 10 as the interval) and report the results in Table 6. For UCMD, $\mathcal{L}_{\text{OSP}}$ achieve the best performance (95.58% mAP) when $T = 45$. For CUB, we can see that the best performance (65.0% R@1 and 23.7% mAP@R) when $T = 65$. It can be observed that the performance is relatively stable on UCMD and CUB in refs. [45, 85].

3) **Effect of relative weight $\omega$**. In Figure 11, we show the effect of relative weight $\omega$ of $\mathcal{L}_{\text{OSAP}}$ on UCMD and CUB respectively. The relative weight $\omega$ in Equation (16) controls the weight between our two training losses $\mathcal{L}_{\text{AP}}^{\text{SRF}}$ and $\mathcal{L}_{\text{OSP}}$: $\omega = 0$ reduces $\mathcal{L}_{\text{OSAP}}$ to $\mathcal{L}_{\text{AP}}^{\text{SRF}}$ while $\omega = 1$ to $\mathcal{L}_{\text{OSP}}$. We can see in Figure 11 that training with the complete $\mathcal{L}_{\text{OSAP}}$ with both $\mathcal{L}_{\text{AP}}^{\text{SRF}}$ and $\mathcal{L}_{\text{OSAP}}$ is always better than using only one of the two losses. On UCMD and CUB datasets, the results both increase by ~2pt in the [0, 0.2] range and then decrease by ~2pt in the [0.2, 1] range. Consequently, the value of 0.2 is the optimal relative weight, achieving the best results on the two datasets.

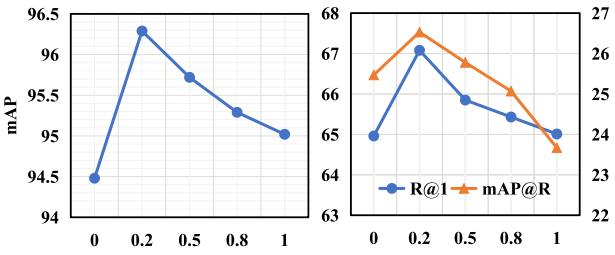
## 5.5 | Comparison with state-of-the-arts

In this section, we compare our OSAP-Loss to the recent AP approximation methods and some relevant state-of-the-art deep metric methods on three RSIR datasets and three CIR datasets. Firstly, we perform a comparison and analysis of AP approximation methods to verify the contribution in our paper. Secondly, we then conduct a further state-of-the-art comparison to demonstrate the superiority and effectiveness of our OSAP-Loss.

**TABLE 6** Ablation study over different hyperparameters: score margin $m$ and temperature parameter $T$

| | UCMD | CUB | | | UCMD | CUB | |
|---|---|---|---|---|---|---|---|
| $m$ | mAP | R@1 | mAP@R | $T$ | mAP | R@1 | mAP@R |
| 0 | **95.45** | **65.1** | 23.6 | 45 | **95.58** | **65.0** | 23.4 |
| 0.15 | 95.07 | 64.9 | **23.7** | 55 | 95.30 | 64.9 | 23.6 |
| 0.25 | 95.05 | 65.0 | **23.7** | 65 | 95.05 | **65.0** | **23.7** |
| 0.35 | 94.98 | 64.6 | **23.7** | 75 | 94.84 | 64.7 | **23.7** |
| 0.45 | 94.89 | 64.6 | **23.7** | 85 | 94.68 | 64.1 | 23.6 |

*Note*: The results are reported on UCMD and CUB. The results in bold indicate the best performance.




(a) relative weight $\omega$ on UCMD (b) relative weight $\omega$ on CUB

**FIGURE 11** Illustration of the effect of relative weight $\omega$ on (a) UCMD and (b) CUB datasets respectively.

**TABLE 7** Comparison between OSAP and state-of-the-art AP approximation methods

| | CUB | | SOP | | INaturalist | |
|---|---|---|---|---|---|---|
| Method | R@1 | mAP@R | R@1 | mAP@R | R@1 | mAP@R |
| FastAP [17] | 58.9 | 22.9 | 78.2 | 51.3 | 53.5 | 19.6 |
| SoftBin [20] | 61.2 | 24.0 | 80.1 | 53.5 | 56.6 | 20.1 |
| BlackBox [24] | 62.6 | 23.9 | 80.0 | 53.1 | 52.3 | 15.2 |
| SmoothAP [25] | 62.1 | 23.9 | 80.9 | 54.6 | 59.8 | 20.7 |
| ROADMAP [85] | 64.2 | 25.3 | **82.0** | **56.5** | 64.5 | 25.1 |
| OSAP (ours) | **67.1** | **26.5** | 78.7 | 51.4 | **66.3** | **25.4** |

| Method | UCMD mAP | NWPUD mAP | PatternNet mAP |
|---|---|---|---|
| FastAP [17] | **96.60** | 94.16 | 99.95 |
| SoftBin [20] | 96.58 | **96.66** | 99.84 |
| BlackBox [24] | 52.15 | 15.33 | 6.93 |
| SmoothAP [25] | 96.49 | 92.53 | 99.93 |
| ROADMAP [85] | 95.76 | 90.35 | 99.84 |
| OSAP (ours) | 96.29 | 92.60 | **99.98** |

*Note*: We report the results on six datasets. The results in bold indicate the best performance.

Abbreviations: AP, average precision; mAP, mean average precision; OSAP, Optimising samples after positive ones & average precision loss; SOP, Stanford Online Product.

**TABLE 8** Comparison with state-of-the-art loss functions on UCMD [40], NWPUD [41] and PatternNet [1]

| | | UCMD | | | | | | NWPUD | | | | | | PatternNet | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Venue** | **mAP** | **P@5** | **P@10** | **P@50** | **P@100** | **P@1000** | **mAP** | **P@5** | **P@10** | **P@50** | **P@100** | **P@1000** | **mAP** | **P@5** | **P@10** | **P@50** | **P@100** | **P@1000** |
| Triplet [75] | ICCV'17 | 92.96 | 98.04 | 96.63 | 92.62 | 46.16 | 4.69 | 93.82 | 98.65 | 96.85 | 96.07 | 94.83 | 15.34 | 94.94 | 99.52 | 97.92 | 96.13 | 95.07 | 15.61 |
| N-pair-mc [69] | NIPS'16 | 91.81 | 94.04 | 91.46 | 90.49 | 45.08 | 4.67 | 93.06 | 97.86 | 95.12 | 94.35 | 98.15 | 15.46 | 94.11 | 97.94 | 95.15 | 94.33 | 98.17 | 15.52 |
| ProxyNCA [70] | ICCV'17 | 95.72 | 97.98 | 96.65 | 94.23 | 47.02 | 4.71 | 97.68 | 97.54 | 97.57 | 97.91 | 97.44 | 15.69 | 97.71 | 98.56 | 98.69 | 98.89 | 98.45 | 15.74 |
| LiftedStruct [43] | CVPR'16 | 96.08 | 98.90 | 97.82 | 95.78 | 47.46 | 4.76 | 97.47 | 97.03 | 97.42 | 97.63 | 97.72 | 15.76 | 98.58 | 98.05 | 98.62 | 98.75 | 98.88 | 15.79 |
| DSLL [71] | Entropy'19 | 97.34 | 98.98 | 98.42 | 96.93 | 48.67 | 4.86 | 98.54 | 99.05 | 98.15 | 96.34 | 98.45 | 15.78 | 98.52 | 99.09 | 98.03 | 96.68 | 98.69 | 15.83 |
| DCL [2] | Remote Sensing'20 | 98.76 | 100.00 | 100.00 | 99.33 | 49.82 | 5.21 | 99.44 | 100.00 | 100.00 | 99.91 | 99.70 | 16.42 | 99.43 | 100.00 | 100.00 | 99.89 | 99.66 | 16.38 |
| RRL [3] | TGRS'20 | 99.52 | 100.00 | 100.00 | 99.80 | – | – | 99.61 | – | – | – | – | – | – | – | – | – | – | – |
| FastAP [17] | CVPR'19 | 96.61 | 100.00 | 100.00 | 99.99 | 78.99 | 7.90 | 94.38 | 95.74 | 95.54 | 95.12 | 94.52 | 13.88 | 99.95 | 99.96 | 99.96 | 99.96 | 99.96 | 55.89 |
| BlackBox [24] | CVPR'20 | 52.70 | 86.40 | 80.52 | 58.98 | 44.25 | 7.80 | 15.35 | 40.65 | 35.25 | 22.53 | 17.45 | 6.24 | 4.17 | 2.63 | 2.63 | 2.63 | 2.63 | 2.63 |
| SmoothAP [25] | ECCV'20 | 96.49 | 99.98 | 99.96 | 99.91 | 78.94 | 7.90 | 92.67 | 95.53 | 95.25 | 94.52 | 93.12 | 13.89 | 99.93 | 99.96 | 99.95 | 99.95 | 99.95 | 55.89 |
| SoftBin* [20] | ICCV'19 | 96.58 | 100.00 | 100.00 | 99.98 | 78.99 | 7.90 | 94.05 | 95.85 | 95.66 | 95.20 | 94.30 | 13.69 | 99.95 | 99.97 | 99.97 | 99.97 | 99.97 | 55.92 |
| ROADMAP [85] | NIPS'21 | 95.94 | 99.73 | 99.71 | 99.47 | 78.51 | 7.90 | 90.37 | 94.50 | 94.14 | 93.00 | 91.52 | 13.80 | 99.91 | 99.96 | 99.96 | 99.95 | 99.94 | 55.88 |
| OSAP (w/o MFM) | This work | 96.47 | 99.88 | 99.89 | 99.86 | 78.94 | 7.90 | 93.55 | 95.34 | 95.14 | 94.49 | 93.56 | 13.90 | 99.98 | 99.98 | 99.98 | 99.98 | 99.98 | 55.90 |
| OSAP (w/MFM) | This work | **99.64** | **100.00** | **100.00** | **99.94** | **79.54** | **7.94** | **99.76** | **100.00** | **100.00** | **99.95** | **99.84** | **13.97** | **99.99** | **10.00** | **100.00** | **100.00** | **100.00** | **56.31** |

*Note*: The embedding size is 2048. We report the results of mAP and P@k. All methods rely on a standard convolutional backbone (generally ResNet-50). The results in bold indicate the best performance.

Abbreviations: mAP, mean average precision; MFM, memory-free mechanism; OSAP, Optimising samples after positive ones & average precision loss.

### 5.5.1 | Comparison to AP approximation methods

Here, we train all models using the same experimental setting (Backbone: ResNet-50; Batch size: 64; Embedding size: 512) as in Section 5.4. And, to compare our OSAP with recently proposed AP approximation methods, we conduct the experiments on three RSIR datasets and three CIR datasets. We report the results in Table 7. Specifically, we compare OSAP on three RSIR datasets and to recent AP approximation losses, including the soft-binning approaches FastAP [17] and SoftBin [20], BlackBox [24], SmoothAP [25], PNP-$D_q$ [84] and ROADMAP [85]. We can see that OSAP outperforms most of the current AP approximation methods by a significant margin. The performance improvement is especially significant on large-scale datasets, such as the PatternNet dataset in RSIR and the INaturalist dataset in CIR.

### 5.5.2 | Comparison with state-of-the-arts

We use the same setting as in Section 5.4. For RSIR datasets, we only change the embedding size to 2048. For CIR datasets, we follow standard practices for ResNet-50 by using larger images (256 × 256 on SOP and CUB) and using max instead of average pooling and layer normalisation for CUB. We compare OSAP-Loss to other state-of-the-art methods across three RSIR datasets and three CIR datasets, and we report the results in Tables 8 and 9 respectively. We divide the comparison methods into two main categories: deep metric learning and AP approximation methods. OSAP-Loss falls in the second category. We show the results of OSAP-Loss with MFM and without MFM separately.

In Table 8, OSAP-Loss (w/o MFM) outperforms all previous methods on PatternNet dataset. We can see that the performances of the methods on the RSIR dataset are close to saturation. In Table 9, OSAP-Loss (w/o MFM) outperforms all previous AP approximation methods on CUB dataset and INaturalist dataset. It can be observed that MFM can increase the performance of OSAP-Loss by a large margin on all datasets, which further demonstrates the effectiveness of MFM.

### 5.5.3 | Visual assessment results

As a qualitative assessment, we show some retrieval results of OSAP-Loss on RSIR datasets in Figure 12. We show the query

**TABLE 9** Comparison of state-of-the-art loss functions on SOP [43], CUB [42], and INaturalist [44]

| | | | | SOP | | | CUB | | | | INaturalist | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Method** | **Venue** | **Dim** | **1** | **10** | **100** | **1** | **2** | **4** | **8** | **1** | **4** | **16** | **32** |
| Deep metric learning | LiftedStruct [43] | CVPR'16 | 512 | 62.1 | 79.8 | 91.3 | 47.2 | 58.9 | 70.2 | 80.2 | – | – | – | – |
| | Margin [75] | ICCV'17 | 512 | 72.7 | 86.2 | 93.8 | 63.6 | 74.4 | 83.1 | 90.0 | 58.1 | 75.5 | 86.8 | 90.7 |
| | ProxyNCA [70] | ICCV'17 | 512 | 73.7 | – | – | 49.2 | 61.9 | 67.9 | 72.4 | 61.6 | 77.4 | 87.0 | 90.6 |
| | MIC [53] | ICCV'19 | 512 | 77.2 | 89.4 | 95.6 | 66.1 | 76.8 | 85.6 | – | – | – | – | – |
| | MS [54] | CVPR'19 | 512 | 78.2 | 90.5 | 96.0 | 65.7 | 77.0 | 86.3 | 91.2 | – | – | – | – |
| | SoftTriple [76] | ICCV'19 | 512 | 78.3 | 90.3 | 95.9 | 65.4 | 76.4 | 84.5 | 90.4 | – | – | – | – |
| | Circle [55] | CVPR'20 | 512 | 78.3 | 90.5 | 96.1 | 66.7 | 77.4 | 86.2 | 91.2 | – | – | – | – |
| | SEC [56] | NIPS'20 | 512 | 78.7 | 90.8 | 96.6 | 68.8 | 79.4 | 87.2 | 92.5 | – | – | – | – |
| | HORDE [57] | ICCV'19 | 512 | 80.1 | 91.3 | 96.2 | 66.8 | 77.4 | 85.1 | 91.0 | – | – | – | – |
| | XBM [91] | CVPR'20 | 128 | 80.6 | 91.6 | 96.2 | 65.8 | 75.9 | 84.0 | 89.9 | – | – | – | – |
| | Triplet SCT [58] | ECCV'20 | 512/64 | 81.9 | 92.6 | 96.8 | 57.7 | 69.8 | 79.6 | 87.0 | – | – | – | – |
| AP-based loss | FastAP [17] | CVPR'19 | 512 | 76.4 | 89.0 | 95.1 | – | – | – | – | 60.6 | 77.0 | 87.2 | 90.6 |
| | BlackBox [24] | CVPR'20 | 512 | 78.6 | 90.5 | 96.0 | 64.0 | 75.3 | 84.1 | 90.6 | 62.9 | 79.4 | 88.7 | 91.7 |
| | SmoothAP [25] | ECCV'20 | 512 | 80.1 | 91.5 | 96.6 | – | – | – | – | 67.2 | 81.8 | 90.3 | 93.1 |
| | SoftBin* [20] | ICCV'19 | 512 | 80.6 | 91.3 | 96.1 | 61.2 | 73.14 | 83.0 | 89.5 | 64.2 | 77.1 | 82.7 | 91.7 |
| | PNP-$D_q$ [84] | AAAI'22 | 512 | 81.1 | 92.7 | 96.3 | – | – | – | – | 66.6 | 81.1 | 89.7 | 92.6 |
| | ROADMAP [85] | NIPS'21 | 512 | 83.1 | 92.2 | 96.8 | 68.5 | 78.7 | 86.6 | 91.9 | 69.1 | 83.1 | 91.3 | 93.9 |
| | OSAP (w/o MFM) | This work | 512 | 79.9 | 91.3 | 96.5 | 69.8 | 79.6 | 88.0 | 92.8 | 70.7 | 84.3 | 92.5 | 94.7 |
| | OSAP (w/MFM) | This work | 512 | **84.4** | **93.1** | **97.3** | **70.5** | **80.2** | **88.3** | **93.2** | **71.0** | **84.7** | **92.8** | **94.9** |

*Note*: The embedding size is 512. We report the results of Recall@k. All methods rely on a standard convolutional backbone (generally ResNet-50). The results in bold indicate the best performance.

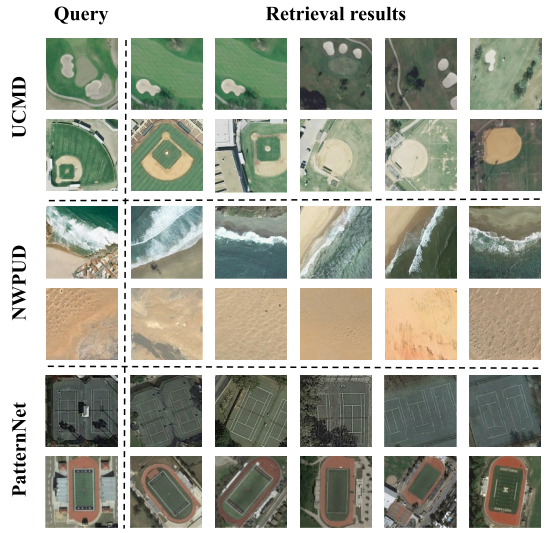



**FIGURE 12** Illustration of image retrieval examples on three RSIR datasets. Golf course and baseball diamond in UCMD dataset (retrieval accuracy-99.64%), beach and desert in NWPUD (retrieval accuracy-99.76%), tennis court and football field in PtternNet dataset (retrieval accuracy-99.99%).

images, the most five similar retrieved images, and the corresponding retrieval accuracy. We can find that the semantic quality of the retrieval results is very good. These results also further demonstrate the effectiveness of our proposed OSAP-Loss.

## 6 | CONCLUSION

In this paper, we introduce a novel OSAP-Loss for RSIR, which improves the retrieval performance by directly optimising AP. Specifically, OSAP-Loss consists of three components: $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$, $\mathcal{L}_{\mathrm{OSP}}$, and MFM. To solve the weakness of sigmoid-based AP approximation, we proposed a SRF to replace the sigmoid-based ranking function, leading to the $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ being accurate. Afterwards, to overcome the non-decomposability in AP optimisation, we equip the $\mathcal{L}_{\mathrm{AP}}^{\mathrm{SRF}}$ with $\mathcal{L}_{\mathrm{OSP}}$ to reduce the decomposability gap. Moreover, we develop an MFM to further thoroughly address the non-decomposability in AP optimisation, which sidesteps the constraint of the GPU memory to use large batch size training on a single GPU. We provide theoretical analysis as well as experimental results to demonstrate the superiority and effectiveness of OSAP-Loss. Extensive experiments that OSAP-Loss show the superiority and competitive performance on three RSIR datasets and three CIR datasets compared to the state-of-the-arts.

Furthermore, our OSAP-Loss is a data-driven method, which has the potential of propagating dataset biases. On some datasets, we find that the performance cannot be remarkable. We find that the performance cannot be exceptionally good on some datasets. OSAP-Loss is on par with or superior to existing loss functions. Future work plans to improve the robustness of our method and reduce the impact of bias in the dataset. We also plan to improve the stability of our approach and enhance its extensibility and applicability.

## ACKNOWLEDGEMENT

This work was supported by the National Nature Science Foundation of China (No. U1803262, 62176191, 62171325), and Nature Science Foundation of Hubei Province (2022CFB018). This research was financially supported by fund from Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System (Wuhan University of Science and Technology) (ZNXX2022001).

## CONFLICT OF INTEREST

The author declares that there is no conflict of interest that could be perceived as prejudicing the impartiality of the research reported.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in UCMD at https://doi.org/10.1145/1869790.1869829, reference number [39]. The data that support the findings of this study are openly available in NWPUD at https://doi.org/10.1109/JPROC.2017.2675998, reference number [40]. The data that support the findings of this study are openly available in PatternNet at https://doi.org/10.1016/j.isprsjprs.2018.01.004, reference number [1]. The data that support the findings of this study are openly available in CUB at https://authors.library.caltech.edu/27452/1/CUB_200_2011.pdf, reference number [41]. The data that support the findings of this study are openly available in SOP at https://doi.org/10.1109/CVPR.2016.434, reference number [42]. The data that support the findings of this study are openly available in INaturalist at https://doi.org/10.1109/CVPR.2018.00914, reference number [43].

## ORCID

*Xin Yuan* https://orcid.org/0000-0003-3140-3243
*Xin Xu* https://orcid.org/0000-0003-0748-3669
*Xiao Wang* https://orcid.org/0000-0003-0770-9891
*Kai Zhang* https://orcid.org/0000-0003-0318-3255
*Liang Liao* https://orcid.org/0000-0002-2238-2420
*Zheng Wang* https://orcid.org/0000-0003-3846-9157
*Chia-Wen Lin* https://orcid.org/0000-0002-9097-2318

## REFERENCES

1. Zhou, W., et al.: Patternnet: a benchmark dataset for performance evaluation of remote sensing image retrieval. ISPRS J. Photogrammetry Remote Sens. 145, 197–209 (2018)
2. Fan, L., Zhao, H., Zhao, H.: Distribution consistency loss for large-scale remote sensing image retrieval. Rem. Sens. 12(1), 175 (2020)
3. Fan, L., Zhao, H., Zhao, H.: Global optimization: combining local loss with result ranking loss in remote sensing image retrieval. IEEE Trans. Geosci. Rem. Sens. 59(8), 7011–7026 (2020)
4. Kang, J., et al.: Robust normalized softmax loss for deep metric learning-based characterization of remote sensing images with label noise. IEEE Trans. Geosci. Rem. Sens. 59(10), 8798–8811 (2021)
5. Ge, W.: Deep metric learning with hierarchical triplet loss. In: Proceedings of the European Conference on Computer Vision, pp. 269–285 (2018)
6. Chakrabarti, S., et al.: Structured learning for non-smooth ranking losses. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 88–96 (2008)

7. Caruana, R., et al.: Ensemble selection from libraries of models. In: Proceedings of the International Conference on Machine Learning, pp. 1–8 (2004)
8. Yue, Y., et al.: A support vector method for optimizing average precision. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 271–278 (2007)
9. Constantin, M.G., et al.: The predicting media memorability task at mediaeval 2019. In: Working Notes Proceedings of the MediaEval 2019 Workshop, pp. 27–29 (2019)
10. Han, L., et al.: Cohesion intensive deep hashing for remote sensing image retrieval. Rem. Sens. 12(1), 101 (2019)
11. Oksuz, K., et al.: Rank & sort loss for object detection and instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3009–3018 (2021)
12. Lin, Y., Lee, Y., Wahba, G.: Support vector machines for classification in nonstandard situations. Mach. Learn. 46(1), 191–202 (2002)
13. Morik, K., Brockhausen, P., Joachims, T.: Combining statistical learning with a knowledge-based approach: a case study in intensive care monitoring. In: Proceedings of the International Conference on Machine Learning, pp. 268–277 (1999)
14. Bartell, B.T., Cottrell, G.W., Belew, R.K.: Automatic combination of multiple ranked retrieval systems. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 173–181 (1994)
15. Herschtal, A., Raskutti, B.: Optimising area under the roc curve using gradient descent. In: Proceedings of the International Conference on Machine Learning, pp. 49 (2004)
16. Bruch, S., et al.: An analysis of the softmax cross entropy loss for learning-to-rank with binary relevance. In: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 75–78 (2019)
17. Cakir, F., et al.: Deep metric learning to rank. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1861–1870 (2019)
18. He, K., et al.: Hashing as tie-aware learning to rank. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4023–4032 (2018)
19. He, K., Lu, Y., Sclaroff, S.: Local descriptors optimized for average precision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 596–605 (2018)
20. Revaud, J., et al.: Learning with average precision: training image retrieval with a listwise loss. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5107–5116 (2019)
21. Ustinova, E., Lempitsky, V.: Learning deep embeddings with histogram loss. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 4177–4185 (2016)
22. Engilberge, M., et al.: Sodeep: a sorting deep net to learn ranking loss surrogates. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10792–10801 (2019)
23. Pogančić, M.V., et al.: Differentiation of blackbox combinatorial solvers. In: Proceedings of the International Conference on Learning Representations, pp. 1–19 (2020)
24. Rolínek, M., et al.: Optimizing rank-based metrics with blackbox differentiation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7620–7630 (2020)
25. Brown, A., et al.: Smooth-ap: smoothing the path towards large-scale image retrieval. In: Proceedings of the European Conference on Computer Vision, pp. 677–694 (2020)
26. Liu, Z., et al.: Image retrieval on real-life images with pre-trained vision-and-language models. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2125–2134 (2021)
27. Sain, A., et al.: Stylemeup: towards style-agnostic sketch-based image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8504–8513 (2021)
28. Li, X., et al.: Qair: practical query-efficient black-box attacks for image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3330–3339 (2021)
29. Tadepalli, Y., et al.: Content-based image retrieval using Gaussian–Hermite moments and firefly and grey wolf optimization. CAAI Trans. Intell. Technol. 6(2), 135–146 (2021)
30. Ahmad, F.: Deep image retrieval using artificial neural network interpolation and indexing based on similarity measurement. CAAI Trans. Intell. Technol. 7(2), 200–218 (2022)
31. Deng, J., et al.: Masked face recognition challenge: the insightface track report. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1437–1444 (2021)
32. Meng, Q., et al.: Magface: a universal representation for face recognition and quality assessment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 14225–14234 (2021)
33. Zhu, Z., et al.: Webface260m: a benchmark unveiling the power of million-scale deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10492–10502 (2021)
34. Liao, L., et al.: Exploiting effects of parts in fine-grained categorization of vehicles. In: Proceedings of the IEEE International Conference on Image Processing, pp. 745–749 (2015)
35. Wang, Z., et al.: Re-identification = retrieval+ verification: back to essence and forward with a new metric. *arXiv preprint*, *arXiv:201111506* (2020)
36. Xie, P., et al.: Unsupervised video person re-identification via noise and hard frame aware clustering. In: 2021 IEEE International Conference on Multimedia and Expo, pp. 1–6 (2021)
37. Jiang, M., et al.: Robust vehicle re-identification via rigid structure prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4026–4033 (2021)
38. Xu, X., et al.: Rank-in-rank loss for person re-identification. ACM Trans. Multimed. Comput. Commun. Appl. 18, 1–21 (2022)
39. Xu, X., et al.: Towards generalizable person re-identification with a bi-stream generative model. Pattern Recogn. 132, 108954 (2022)
40. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 270–279 (2010)
41. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: benchmark and state of the art. Proc. IEEE 105(10), 1865–1883 (2017)
42. Wah, C., et al.: The caltech-ucsd birds-200-2011 dataset (2011)
43. OhSong, H., et al.: Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4004–4012 (2016)
44. Van Horn, G., et al.: The inaturalist species classification and detection dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8769–8778 (2018)
45. Cao, Z., et al.: Hashnet: deep learning to hash by continuation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5608–5617 (2017)
46. Zhang, Z., et al.: Probability ordinal-preserving semantic hashing for large-scale image retrieval. ACM Trans. Knowl. Discov. Data 15(3), 1–22 (2021)
47. Weng, Z., Zhu, Y.: Online hashing with bit selection for image retrieval. IEEE Trans. Multimed. 23, 1868–1881 (2021)
48. Shen, F., et al.: Unsupervised deep hashing with similarity-adaptive and discrete optimization. IEEE Trans. Pattern Anal. Mach. Intell. 40(12), 3034–3044 (2018)
49. Luo, Y., et al.: Robust discrete code modeling for supervised hashing. Pattern Recogn. 75, 128–135 (2018)
50. Perronnin, F., et al.: Large-scale image retrieval with compressed Fisher vectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3384–3391 (2010)
51. Arandjelovic, R., Zisserman, A.: All about VLAD. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1578–1585 (2013)
52. Husain, S.S., Ong, E.J., Bober, M.: Actnet: end-to-end learning of feature activations and multi-stream aggregation for effective instance image retrieval. Int. J. Comput. Vis. 129(5), 1432–1450 (2021)
53. Roth, K., Brattoli, B., Ommer, B.: Mic: mining interclass characteristics for improved metric learning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 8000–8009 (2019)

54. Wang, X., et al.: Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5022–5030 (2019)
55. Sun, Y., et al.: Circle loss: a unified perspective of pair similarity optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6398–6407 (2020)
56. Zhang, D., Li, Y., Zhang, Z.: Deep metric learning with spherical embedding. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 18772–18783 (2020)
57. Jacob, P., et al.: Metric learning with horde: high-order regularizer for deep embeddings. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6539–6548 (2019)
58. Xuan, H., et al.: Hard negative examples are hard, but useful. In: Proceedings of the European Conference on Computer Vision, pp. 126–142 (2020)
59. Zhao, H., et al.: Global-aware ranking deep metric learning for remote sensing image retrieval. Geosci. Rem. Sens. Lett. IEEE 19, 1–5 (2021)
60. Kang, J., et al.: Rotation-invariant deep embedding for remote sensing images. IEEE Trans. Geosci. Rem. Sens. 60, 1–13 (2021)
61. Gu, M., et al.: Polsar ship detection based on a sift-like polsar keypoint detector. Rem. Sens. 14(12), 2900 (2022)
62. Mikriukov, G., Ravanbakhsh, M., Demir, B.: Deep unsupervised contrastive hashing for large-scale cross-modal text-image retrieval in remote sensing. *arXiv preprint*, *arXiv:220108125* (2022)
63. Aptoula, E.: Remote sensing image retrieval with global morphological texture descriptors. IEEE Trans. Geosci. Rem. Sens. 52(5), 3023–3034 (2014)
64. Scott, G.J., et al.: Entropy-balanced bitmap tree for shape-based object retrieval from large-scale satellite imagery databases. IEEE Trans. Geosci. Rem. Sens. 49(5), 1603–1616 (2011)
65. Hua, Y., et al.: Semantic segmentation of remote sensing images with sparse annotations. Geosci. Rem. Sens. Lett. IEEE 19, 1–5 (2021)
66. Zhou, W., et al.: Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval. Rem. Sens. 9(5), 489 (2017)
67. Radenović, F., Tolias, G., Chum, O.: CNN image retrieval learns from bow: unsupervised fine-tuning with hard examples. In: Proceedings of the European Conference on Computer Vision, pp. 3–20 (2016)
68. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
69. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 1857–1865 (2016)
70. Movshovitz-Attias, Y., et al.: No fuss distance metric learning using proxies. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 360–368 (2017)
71. Fan, L., et al.: Distribution structure learning loss (DSLL) based on deep metric learning for image retrieval. Entropy 21(11), 1121 (2019)
72. Koestinger, M., et al.: Large scale metric learning from equivalence constraints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2288–2295 (2012)
73. Swets, D.L., Weng, J.J.: Using discriminant eigenfeatures for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 18(8), 831–836 (1996)
74. Harwood, B., et al.: Smart mining for deep metric learning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2821–2829 (2017)
75. Wu, C.Y., et al.: Sampling matters in deep embedding learning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2840–2848 (2017)
76. Qian, Q., et al.: Softtriple loss: deep metric learning without triplet sampling. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6450–6458 (2019)
77. Teh, E.W., DeVries, T., Taylor, G.W.: Proxynca++: revisiting and revitalizing proxy neighborhood component analysis. In: Proceedings of the European Conference on Computer Vision, pp. 448–464 (2020)
78. Wang, X., et al.: Ranked list loss for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5207–5216 (2019)
79. Musgrave, K., Belongie, S., Lim, S.N.: A metric learning reality check. In: Proceedings of the European Conference on Computer Vision, pp. 681–699 (2020)
80. Kim, S., et al.: Proxy anchor loss for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3238–3247 (2020)
81. Durand, T., Thome, N., Cord, M.: Exploiting negative evidence for deep latent structured models. IEEE Trans. Pattern Anal. Mach. Intell. 41(2), 337–351 (2018)
82. McFee, B., Lanckriet, G.R.: Metric learning to rank. In: Proceedings of the International Conference on Machine Learning, pp. 1–8 (2010)
83. Mohapatra, P., et al.: Efficient optimization for rank-based loss functions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3693–3701 (2018)
84. Li, Z., et al.: Rethinking the optimization of average precision: only penalizing negative instances before positive ones is enough. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1518–1526 (2022)
85. Ramzi, E., et al.: Robust and decomposable average precision for image retrieval. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 23569–23581 (2021)
86. Law, M.T., Thome, N., Cord, M.: Learning a distance metric from relative comparisons between quadruplets of images. Int. J. Comput. Vis. 121(1), 65–94 (2017)
87. Mohapatra, P., Jawahar, C., Kumar, M.P.: Efficient optimization for average precision SVM. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 2312–2320 (2014)
88. Suh, Y., et al.: Stochastic class-based hard example mining for deep metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7251–7259 (2019)
89. Carvalho, M., et al.: Cross-modal retrieval in the cooking context: learning semantic text-image embeddings. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 35–44 (2018)
90. Faghri, F., et al.: Vse++: improving visual-semantic embeddings with hard negatives. *arXiv preprint*, *arXiv:170705612* (2017)
91. Wang, X., et al.: Cross-batch memory for embedding learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6388–6397 (2020)
92. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
93. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 539–546 (2005)
94. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1735–1742 (2006)
95. Ye, F., et al.: Remote sensing image retrieval using convolutional neural network features and weighted distance. Geosci. Rem. Sens. Lett. IEEE 15(10), 1535–1539 (2018)
96. Deng, J., et al.: Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
97. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint*, *arXiv:14126980* (2014)


**How to cite this article:** Yuan, X., et al.: OSAP-Loss: Efficient optimization of average precision via involving samples after positive ones towards remote sensing image retrieval. CAAI Trans. Intell. Technol. 1–22 (2023). https://doi.org/10.1049/cit2.12151