Robust and Decomposable Average Precision for Image Retrieval

Anonymous Author(s)
Affiliation
Address
email

Abstract

In image retrieval, standard evaluation metrics rely on score ranking, e.g. average precision (AP). In this paper, we introduce a method for robust and decomposable average precision (ROADMAP) addressing two major challenges for end-to-end training of deep neural networks with AP: non-differentiability and non-decomposability. Firstly, we propose a new differentiable approximation of the rank function, which provides an upper bound of the AP loss and ensures robust training. Secondly, we design a simple yet effective loss function to reduce the decomposability gap between the AP in the whole training set and its averaged batch approximation, for which we provide theoretical guarantees. Extensive experiments conducted on three image retrieval datasets show that ROADMAP outperforms several recent AP approximation methods and highlight the importance of our two contributions. Finally, using ROADMAP for training deep models yields very good performances, outperforming state-of-the-art results on the three datasets.

1 Introduction

The task of 'query by example' is a major prediction problem, which consists in learning a similarity function able to properly rank all the instances in a retrieval set according to their relevance to the query, such that relevant items have the largest similarity. In computer vision, it drives several major applications, *e.g.* content-based image retrieval, face recognition or person re-identification.

Such tasks are usually evaluated with rank-based metrics, *e.g.* Recall@k, Normalized Discounted Cumulative Gain (NDCG), and Average Precision (AP). AP is also the *de facto* metric used in several vision tasks implying a large imbalance between positive and negative samples, *e.g.* object detection.

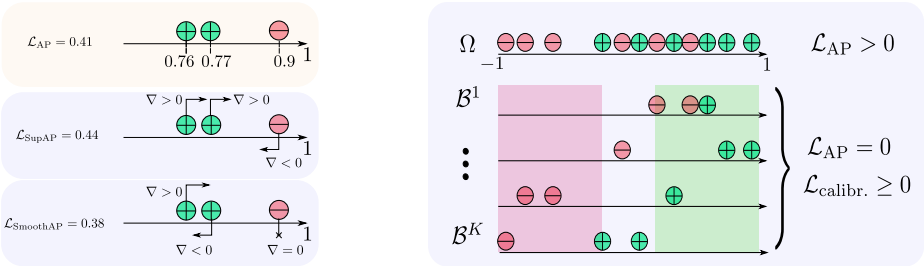
In this paper, we address the problem of direct AP training with stochastic gradient-based optimization, *e.g.* using deep neural networks, which poses two major challenges.

Firstly, the AP loss $\mathcal{L}_{\text{AP}} = 1 - \text{AP}$ is not differentiable and is thus not directly amenable to gradient-based optimization. There has been a rich literature for providing smooth and upper bound surrogate losses for $\mathcal{L}_{\text{AP}}$ [42, 21, 22, 6, 25]. More recently, smooth differentiable rank approximations have been proposed [35, 14, 15, 3, 27, 8, 2], but generally lose the important $\mathcal{L}_{\text{AP}}$ upper bound property.

The second important issue of AP optimization relates to its non-decomposability: $\mathcal{L}_{\text{AP}}^B$ averaged over batches underestimates $\mathcal{L}_{\text{AP}}$ on the whole training dataset, which we refer as the *decomposability gap*. In image retrieval, the attempts to circumvent the problem involve *ad hoc* methods based on batch sampling strategies [10, 32, 20, 32, 30], or storing all training representations/scores [38, 3, 27, 25], leading to complex models with a large computation and memory overhead.

In this paper, we introduce a method for RObust And DecoMposable Average Precision (ROADMAP), which explicitly addresses the aforementioned challenges of AP optimization.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) $\mathcal{L}_{\text{SupAP}} \geq \mathcal{L}_{\text{AP}}$ and $\nabla \mathcal{L}_{\text{SupAP}} > 0$ in this example, in contrast to SmoothAP [2]. This ensures robust training and comes from a new approximation of the rank function.

(b) $\mathcal{L}_{\text{AP}}$ non-decomposability: $\mathcal{L}_{\text{AP}} = 0$ in all batches $\mathcal{B}^i$ despite $\mathcal{L}_{\text{AP}} \neq 0$ over the whole $\bigcup_i \mathcal{B}^i$. $\mathcal{L}_{\text{calibr.}}$ controls the absolute scores between batches, such that $\mathcal{L}_{\text{ROADMAP}} \neq 0$ in each batch.

Figure 1: Our robust and decomposable Average Precision training (ROADMAP) includes (a) a smooth loss $\mathcal{L}_{\text{SupAP}}$ upper-bounding $\mathcal{L}_{\text{AP}}$, and (b) a calibration loss $\mathcal{L}_{\text{calibr.}}$ supporting decomposability.

Our first contribution is to propose a new surrogate loss $\mathcal{L}_{\text{SupAP}}$ for $\mathcal{L}_{\text{AP}}$. In particular, we introduce a smooth approximation of the rank function, with a different behaviour for positive and negative examples. By this design, $\mathcal{L}_{\text{SupAP}}$ provides an upper bound of $\mathcal{L}_{\text{AP}}$, and always back-propagates gradients when the correct ranking is not satisfied. These two features illustrated in the the toy example on Figure 1a are not fulfilled by binning approaches [3, 27] or by SmoothAP [2].

As a second contribution, we propose to improve the non-decomposability in AP training. To this end, we introduce a simple yet effective training objective $\mathcal{L}_{\text{calibr.}}$, which calibrates the scores among different batches by controlling the absolute value of positive and negative samples. We provide a theoretical analysis showing that $\mathcal{L}_{\text{calibr.}}$ decreases the decomposability gap. Figure 1 illustrates how $\mathcal{L}_{\text{calibr.}}$ can be leveraged to improve the overall ranking.

We provide a thorough experimental validation including three standard image retrieval datasets and show that ROADMAP outperforms state-of-the-art methods. We also report the large and consistent gain compared to rank/AP approximation baselines, and we highlight in the ablation studies the importance of our two contributions. Finally, ROADMAP does not entail any memory or computation overhead and remains competitive even with small batches.

2 Related work

We discuss here the literature in image retrieval dedicated to AP optimization, and compare to other approaches based on optimizing representations [23, 1, 43, 45, 33] in the experiments.

Smooth AP approximations Studying smooth surrogate losses for AP has a long history. The widely used surrogate for retrieval is to consider constraints based on pairs [40, 12, 26], triplets [11], quadruplets [18] or n-uplets [30] to enforce partial ranking. These metric learning methods optimize a very coarse upper bound on AP and need complex post-processing and tricks to be effective.

One option for training with AP is to design smooth upper bounds on the AP loss. Seminal works are based on structural SVMs [42, 21], with extensions to speed-up the "loss-augmented inference" [22] or to adapt to weak supervision [6]. Recently, a generic blackbox combinatorial solver has been introduced [25] and applied to AP optimization [28]. To overcome the brittleness of AP with respect to small score variations, an *ad hoc* perturbation is applied to positive and negative scores during training. These methods provide elegant AP upper bounds, but generally are coarse AP approximations.

Other approaches rely on designing smooth approximations of the the rank function. This is done in soft-binning techniques [14, 15, 35, 3, 27] by using a smoothed discretization of similarity scores. Other approaches rely on explicitly approximating the non-differentiable rank functions using neural networks [8], or with a sum of sigmoid functions in the recent SmoothAP approach [2]. These approaches enable accurate AP approximations by providing tight and smooth approximations of the rank function. However, they do not guarantee that the resulting loss is an AP loss upper bound. The $\mathcal{L}_{\text{SupAP}}$ introduced in this work is based on a smooth approximation of the rank function leading to an upper bound on the AP loss, making our approach both accurate and robust.

Decomposability in AP optimization Batch training is mandatory in deep learning. However, the non-decomposability of AP is a severe issue, since it yields an inconsistent AP gradient estimator.

Non-decomposability is related to sampling informative constraints in simple AP surrogates, *e.g.* triplet losses, since the constraints' cardinality on the whole training set is prohibitive. This has been addressed by efficient batch sampling [13, 10, 32] or selecting informative constraints within mini-batches [30, 9, 4, 32]. In cross-batch memory technique [38], the authors assume a slow drift in learned representations to store them and compute global mining in pair-based deep metric learning.

In AP optimization, the non-decomposability has essentially been addressed by a brute force increase of the batch size [3, 27, 25]. This includes an important overhead in computation and memory, generally involving a two-step approach for first computing the AP loss and subsequently re-computing activations and back-propagating gradients. In contrast, our loss $\mathcal{L}_{\text{calibr.}}$ does not add any overhead and enables good performances for AP optimization even with small batches.

3 Robust and decomposable AP training

We present here our method for RObust And DecoMposable AP (ROADMAP) dedicated to direct optimization of a smooth surrogate of AP with stochastic gradient descent (SGD), see Fig. 2.

Training context Let us consider a retrieval set $\Omega = \{\boldsymbol{x_j}\}_{j \in [\![1;N]\!]}$ composed of N elements, and a set of M queries included in Ω, *i.e.* $\mathcal{Q} = \{\boldsymbol{q_i}\}_{i \in [\![1;M]\!]} \subseteq \Omega$. For each query $\boldsymbol{q_i}$, each element in Ω is assigned a label $y(\boldsymbol{x_j}, \boldsymbol{q_i}) \in \{+1; -1\}$, such that $y(\boldsymbol{x_j}, \boldsymbol{q_i}) = 1$ (resp. $y(\boldsymbol{x_j}, \boldsymbol{q_i}) = -1$) if $\boldsymbol{x_j}$ is relevant (resp. irrelevant) with respect to $\boldsymbol{q_i}$. This defines a query-dependent partitioning of Ω such that $\Omega = \mathcal{P}_i \cup \mathcal{N}_i$, where $\mathcal{P}_i := \{\boldsymbol{x_j} \in \Omega | y(\boldsymbol{x_j}, \boldsymbol{q_i}) = +1\}$ and $\mathcal{N}_i := \{\boldsymbol{x_j} \in \Omega | y(\boldsymbol{x_j}, \boldsymbol{q_i}) = -1\}$.

For each $\boldsymbol{x_j} \in \Omega$, we define a prediction model parametrized by parameters $\boldsymbol{\theta}$, *e.g.* a deep neural network, which provides a vectorial embedding $\mathbf{v}_{\boldsymbol{q_i}} \in \mathbb{R}^d$ of each element, *i.e.*: $\mathbf{v}_{\boldsymbol{q_i}} := f_{\boldsymbol{\theta}}(\boldsymbol{q_i})$. In the embedded space $\mathbb{R}^d$, we compute a similarity score between each query $\boldsymbol{q_i}$ and each element in Ω, *e.g.* by using the cosine similarity: $s(\boldsymbol{q_i}, \boldsymbol{x_j}) = \frac{\mathbf{v}_{\mathbf{q_i}}{}^T \mathbf{v_j}}{||\mathbf{v_{q_i}}||^2 ||\mathbf{v_j}||^2}$.

During training, our goal is to optimize, for each query $\boldsymbol{q_i}$, the model parameters $\boldsymbol{\theta}$ such that positive elements are ranked before negatives. More precisely, we aim at minimizing the AP loss $\mathcal{L}_{\text{AP}_i}$ for each query $\boldsymbol{q_i}$ in the retrieval set Ω. Our overall AP loss $\mathcal{L}_{\text{AP}}$ is averaged over all queries:

$$\mathcal{L}_{\text{AP}}(\boldsymbol{\theta}) = 1 - \frac{1}{M} \sum_{i=1}^{M} \text{AP}_i(\boldsymbol{\theta}), \quad \text{AP}_i(\boldsymbol{\theta}) = \frac{1}{|\mathcal{P}_i|} \sum_{k \in \mathcal{P}_i} \text{Pre}(k, \theta) = \frac{1}{|\mathcal{P}_i|} \sum_{k \in \mathcal{P}_i} \frac{\text{rank}^+(k, \theta)}{\text{rank}(k, \theta)} \tag{1}$$

where $\text{Pre}(k, \theta)$ is the precision for the kth positive example $\boldsymbol{x_k}$, $\text{rank}^+(k, \theta)$ its rank among positives $\mathcal{P}_i$, and the $\text{rank}(k, \theta)$ its rank over $\Omega = \mathcal{P}_i \cup \mathcal{N}_i$.

As previously mentioned, there are two main challenges with SGD optimization of AP in Eq. (1): i) $\text{AP}(\boldsymbol{\theta})$ is not differentiable with respect to $\boldsymbol{\theta}$, and ii) AP does not linearly decompose into batches. ROADMAP addresses both issues: we introduce the robust differentiable $\mathcal{L}_{\text{SupAP}}$ surrogate (Section 3.1), and add the $\mathcal{L}_{\text{calibr.}}$ loss (Section 3.2) to improve AP decomposability. Our final loss $\mathcal{L}_{\text{ROADMAP}}$ is a linear combination of $\mathcal{L}_{\text{SupAP}}$ and $\mathcal{L}_{\text{calibr.}}$, weighted by the hyperparameter λ:

$$\mathcal{L}_{\text{ROADMAP}}(\boldsymbol{\theta}) = (1 - \lambda) \cdot \mathcal{L}_{\text{SupAP}}(\boldsymbol{\theta}) + \lambda \cdot \mathcal{L}_{\text{calibr.}}(\boldsymbol{\theta}) \tag{2}$$

3.1 Robustness in smooth rank approximation

The non-differentiablity in Eq (1) comes from the ranking operator, which can be viewed as counting the number of instances that have a similarity score greater than the considered instance, *i.e.*[1]:

$$\text{rank}^+(k) = 1 + \sum_{j \in \mathcal{P}_i \setminus \{k\}} H(s_j - s_k), \quad \text{where } H(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{rank}(k) = \text{rank}^+(k) + \sum_{j \in \mathcal{N}_i} H(s_j - s_k) = \text{rank}^+(k) + \text{rank}^-(k) \tag{3}$$

[1]For the sake of readability we drop in the following the dependence on $\boldsymbol{\theta}$ for the rank, *i.e.* $\text{rank}(k) := \text{rank}(k, \theta)$ and on the query for the similarity, *i.e.* $s_j := s(q_i, x_j)$.

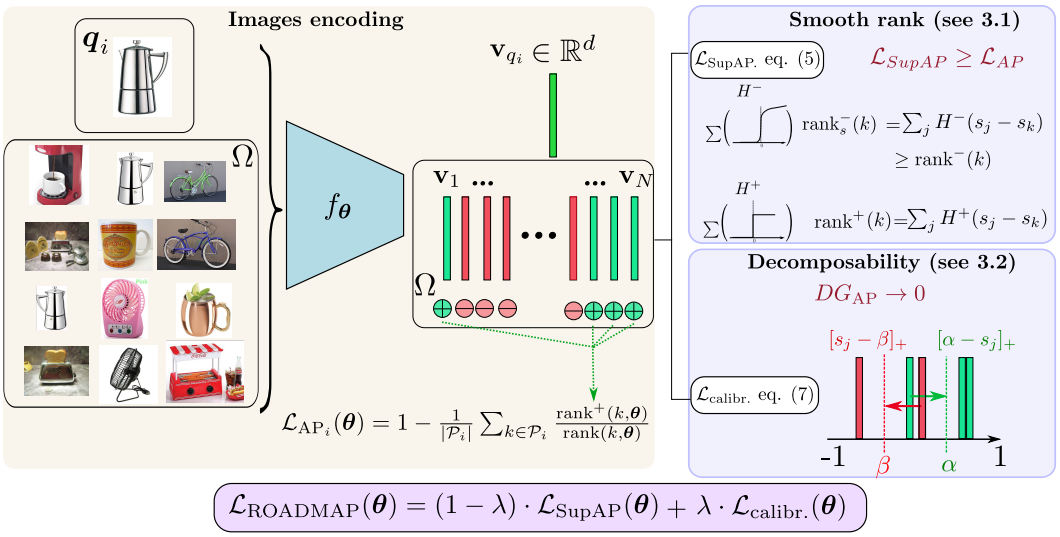


Figure 2: ROADMAP training: we optimize parameters $\boldsymbol{\theta}$ of a deep neural networks to minimize a smooth surrogate of $\mathcal{L}_{\mathrm{AP}_i}(\boldsymbol{\theta})$ between the query $\boldsymbol{q_i}$ and the retrieval set Ω. Our smooth rank approximations H^+ and H^- enables $\mathcal{L}_{\mathrm{SupAP}}$ to be both accurate and robust (sec 3.1), and $\mathcal{L}_{\mathrm{calibr.}}$ enables an implicit batch scores comparison for better decomposability without additional storing (sec 3.2).

From Eq. (3) it becomes clear that the non-differentiablity is due to the Heaviside (step) function H, whose derivative is either zero or undefined. Note that the computation of $\mathrm{rank}^+(k)$ and $\mathrm{rank}^-(k)$ in Eq. (3) relates to the rank of positive instances $\boldsymbol{x_k} \in \mathcal{P}_i$: the score s_k in Eq. (3) is always the score of a positive, whereas s_j can either be a negative's or positive's score.

Smooth loss $\mathcal{L}_{\mathbf{SupAP}}$ To provide a smooth approximation of $\mathcal{L}_{\mathrm{AP}}$ in Eq. (1), we introduce a smooth approximation of the rank function. In particular, we propose a different behaviour between $\mathrm{rank}^+(k)$ and $\mathrm{rank}^-(k)$ in Eq. (3) by defining two functions H^+ and H^-.

For $\mathrm{rank}^+(k)$, we choose to keep the Heaviside (step) function, *i.e.* $H^+ = H$ (see Fig. 3a), which consists in ignoring $\mathrm{rank}^+(k)$ in gradient-based AP optimization. This is done on purpose since $\frac{\partial AP}{\partial\,\mathrm{rank}^+(k)} = \frac{\mathrm{rank}^-(k)}{(\mathrm{rank}^+(k)+\mathrm{rank}^-(k))^2} \geq 0$: the gradient would tend to increase $\mathrm{rank}^+(k)$ and to decrease the score of s_k. Reminding $\boldsymbol{x_k}$ is always a positive instance, this behaviour is undesirable.

For $\mathrm{rank}^-(k)$, we define the following smooth surrogate H^- for H, shown in Fig 3b:

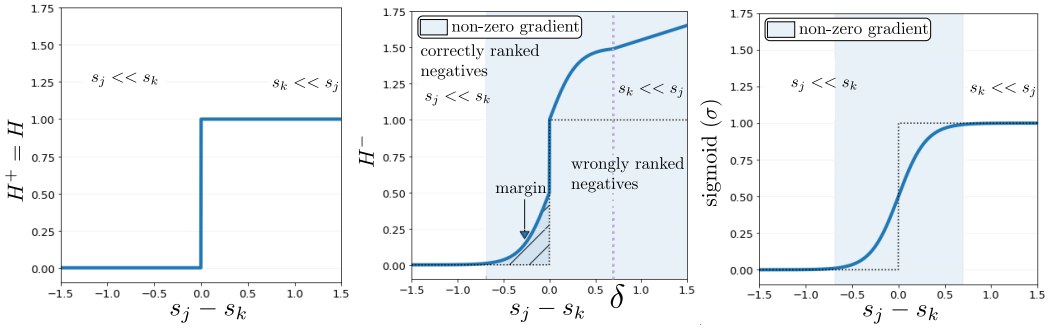
$$H^-(t) = \begin{cases} \sigma(\frac{t}{\tau}) & \text{if } t \leq 0, \quad \text{where } \sigma \text{ is the sigmoid function (Fig. 3c)} \\ \sigma(\frac{t}{\tau}) + 0.5 & \text{if } t \in [0;\delta] \quad \text{with } \delta \geq 0 \\ \rho\cdot(t-\delta) + \sigma(\frac{\delta}{\tau}) + 0.5 & \text{if } t > \delta \end{cases} \tag{4}$$

where τ and ρ are hyperparameters, and δ is defined such that the sigmoidal part of H^- reaches the saturation regime and is fixed for the rest of the paper (see supplementary Sec. A). From the H^- smooth approximation defined in Eq. (4), we obtain the following smooth approximation $\mathrm{rank}_s^-(k) = \sum_{j\in\mathcal{N}_i} H^-(s_j - s_k)$, leading to the following smooth AP loss approximation:

$$\mathcal{L}_{\mathrm{SupAP}}(\boldsymbol{\theta}) = 1 - \frac{1}{M}\sum_{i=1}^{M}\frac{1}{|\mathcal{P}_i|}\sum_{k\in\mathcal{P}_i}\frac{\mathrm{rank}^+(k)}{\mathrm{rank}^+(k) + \mathrm{rank}_s^-(k)} \tag{5}$$

$\mathcal{L}_{\mathbf{SupAP}}$ in Eq. (5) fulfills two main features for AP optimization:

► **(1) $\mathcal{L}_{\mathbf{SupAP}}$ is an upper bound of $\mathcal{L}_{\mathbf{AP}}$ in Eq. (1).** Since H^- in Eq. (4) is an upper bound of a step function (Fig 3b), it is easy to see that $\mathcal{L}_{SupAP} \geq \mathcal{L}_{\mathrm{AP}}$. This is a very important property, since it ensures that the model keeps training until the correct ranking is obtained. It is worth noting that existing smooth rank approximations in the literature [35, 3, 27, 2] do not fulfill this property.



(a) $H^+(x) = H(x)$ in Eq. (3) (b) $H^-(x)$ in Eq. (4) (c) Sigmoid used in [2]

Figure 3: Proposed surrogate losses for the Heaviside (step): with $H^+(x)$ in Fig. 3a and $H^-(x)$ in Fig. 3b, $\mathcal{L}_{\text{SupAP}}$ in Eq. (5) is an upper bound of $\mathcal{L}_{\text{AP}}$. In addition, $H^-(x)$ back-propagates gradients until the correct ranking is satisfied, in contrast to the sigmoid used in [2] (Fig. 3c).

► (2) **$\mathcal{L}_{\text{SupAP}}$ brings training gradients until the correct ranking plus a margin is fulfilled.** When the ranking is incorrect, the negative $\boldsymbol{x_j}$ is ranked before the positive $\boldsymbol{x_k}$, thus $s_j > s_k$ and $H^-(s_j - s_k)$ in Eq. (4) has a non-null derivative. We use a sigmoid to have a large gradient when $s_j - s_k$ is small. To overcome vanishing gradients of the sigmoid for large values $s_j - s_k$, we use a linear function ensuring constant ρ derivative. When the ranking is correct ($s_j < s_k$), we enforce robustness by imposing a margin parametrized by τ (sigmoid in Eq. (4)). This margin overcomes the brittleness of rank losses, which vanish as soon as the ranking is correct [14, 3, 25].

Comparison to SmoothAP [2] $\mathcal{L}_{\text{SupAP}}$ differs from $\mathcal{L}_{\text{SmoothAP}}$ in [2] by i) providing an upper bound on $\mathcal{L}_{\text{AP}}$, ii) improving the gradient flow (Fig. 3b vs Fig. 3c), and iii) overcoming adverse effects of the sigmoid for $rank^+$, as shown in Fig. 1a (and in supplementary sec. A). We experimentally verify the consistent gain brought out by $\mathcal{L}_{\text{SupAP}}$ over $\mathcal{L}_{\text{SmoothAP}}$.

3.2 Decomposable Average Precision

In Eq. (1), AP decomposes linearly between queries $\boldsymbol{q_i}$, but AP_i does not decomposes linearly between samples. We therefore focus our analysis of the non-decomposability on a single query. For a retrieval set Ω of N elements, we consider $\{\mathcal{B}^b\}_{b\in\{1:K\}}$ batches of size B, such that $N/B = K \in \mathbb{N}$. Let $\text{AP}_i^b(\boldsymbol{\theta})$ be the AP in batch b for query $\boldsymbol{q_i}$, we define the "decomposability gap" DG_{AP} as follows:

$$DG_{\text{AP}}(\boldsymbol{\theta}) = \frac{1}{K}\sum_{b=1}^{K} \text{AP}_i^b(\boldsymbol{\theta}) - \text{AP}_i(\boldsymbol{\theta}) \tag{6}$$

$DG_{\text{AP}} \geq 0$ in Eq. (6) is a direct measure of the non-decomposability of AP (see supplementary Sec. A). Our motivation here is to decrease DG_{AP}, *i.e.* to have the average AP over the batches as close as possible to the AP computed over the whole training set. To this aim, we introduce the following loss during training:

$$\mathcal{L}_{\text{calibr.}}(\boldsymbol{\theta}) = \frac{1}{M}\sum_{i=1}^{M} \underbrace{\frac{1}{|\mathcal{P}_i|}\sum_{\boldsymbol{x_j}\in\mathcal{P}_i}[\alpha - s_j]_+}_{\mathcal{L}^+_{\text{calibr.}}} + \underbrace{\frac{1}{|\mathcal{N}_i|}\sum_{\boldsymbol{x_j}\in\mathcal{N}_i}[s_j - \beta]_+}_{\mathcal{L}^-_{\text{calibr.}}} \tag{7}$$

where $[x]_+ = \max(0, x)$. The loss $\mathcal{L}^+_{\text{calibr.}}$ enforces the score of the positive $\boldsymbol{x_i} \in \mathcal{P}_i$ to be larger than α, and $\mathcal{L}^-_{\text{calibr.}}$ enforces the score of the negative $\boldsymbol{x_j} \in \mathcal{N}_i$ to be smaller than $\beta < \alpha$. $\mathcal{L}_{\text{calibr.}}$ is a standard pair-based loss [12], which we revisit in our context to "calibrate" the values of the scores between mini-batches: intuitively, the fact that the positive (resp. negative) scores are above (resp. below) a threshold in the mini-batches makes the average AP closer to the AP on the whole dataset.

Upper bound on the decomposabilty gap To formalize this idea, we provide a theoretical analysis of the impact on the global ranking of $\mathcal{L}_{\text{calibr.}}$ in Eq. (7). Firstly, we can see that if $\mathcal{L}^-_{\text{calibr.}} = \mathcal{L}^+_{\text{calibr.}} = 0$, on each batch, the overall AP and the AP in batches is null, *i.e.* $DG_{\text{AP}}(\boldsymbol{\theta}) = 0$ and we get a

decomposable AP. In a more general setting, we show that minimizing $\mathcal{L}_{\text{calibr.}}$ on each batch reduces the decomposability gap, hence improving the decomposability of the AP.

Let's consider K batches $\{\mathcal{B}^b\}_{b\in\{1:K\}}$ of batch size B divided in $\mathcal{P}_i^b$ positive instances and $\mathcal{N}_i^b$ negative instances w.r.t. the query $\boldsymbol{q_i}$. To give some insight we assume that the AP of each batch is one (*i.e.* $AP_i^b = 1$), and give the following upper bound of DG_{AP} :

$$0 \leq DG_{\text{AP}} \leq 1 - \frac{1}{\sum_{b=1}^{K} |\mathcal{P}_i^b|} \left(\sum_{b=1}^{K} \sum_{j=1}^{B} \frac{j + |\mathcal{P}_i^1| + \cdots + |\mathcal{P}_i^{b-1}|}{j + |\mathcal{P}_i^1| + \cdots + |\mathcal{P}_i^{b-1}| + |\mathcal{N}_i^1| + \cdots + |\mathcal{N}_i^{b-1}|} \right) \quad (8)$$

This upper bound of the decomposability gap is given in the worst case for the global AP : the global ranking is built from the juxtaposition of the batches (see supplementary Sec. A).

We can refine this upper bound by introducing the calibration loss $\mathcal{L}_{\text{calibr.}}$ and constraining the scores of positive and negative instances to be well calibrated. On each batch we define the following quantities $E_b^- = \sum_{j\in\mathcal{N}_i^-} \mathbb{1}(s_j > \beta)$ which are the negative instances that do not respect the constraints and $G_b^- = \sum_{j\in\mathcal{N}_i^-} \mathbb{1}(s_j \leq \beta)$ the negative instances that do. We similarly define E_b^+ and G_b^+. We then have the following upper bound on the decomposability gap :

$$0 \leq DG_{\text{AP}} \leq 1 - \frac{1}{\sum_{b=1}^{K} |\mathcal{P}_i^b|} \left(\sum_{b=1}^{K} \left[\sum_{j=1}^{G_b^+} \frac{j + G_1^+ + \cdots + G_{b-1}^+}{j + G_1^+ + \cdots + G_{b-1}^+ + E_1^- + \ldots E_{b-1}^-} + \right.\right. \quad (9)$$
$$\left.\left. \sum_{j=1}^{E_b^+} \frac{j + G_b^+ + |\mathcal{P}_i^1| + \cdots + |\mathcal{P}_i^{b-1}|}{j + G_b^+ + |\mathcal{P}_i^1| + \cdots + |\mathcal{P}_i^{b-1}| + |\mathcal{N}_i^1| + \cdots + |\mathcal{N}_i^{b-1}|} \right] \right)$$

This refined upper bound is tighter than the upper bound of Eq. (8). Our new $\mathcal{L}_{\text{calibr.}}$ loss directly optimizes this upper bound, making it tighter, hence improving the decomposability of the AP (see supplementary Sec. A).

4 Experiments

Experimental setup We evaluate ROADMAP on the following three image retrieval datasets:
CUB-200-2011 [36] contains 11 788 images of birds classified into 200 fine-grained classes. We follow the standard protocol and use the first (resp. last) 100 classes for training (resp. evaluation).
Stanford Online Product (SOP) [31] is a dataset with 120 053 images of 22 634 objects classified into 12 categories (*e.g.* bikes, coffee makers). We use the reference train and test splits from [31].
INaturalist-2018 is a large scale dataset of 461 939 wildlife animals images classified into 8142 classes. We use the splits from [2] with 70% of the classes in the train set and the rest in the test set.

ROADMAP settings For all experiments in Section 4.1 and Section 4.2, we use $\lambda = 0.5$ for $\mathcal{L}_{\text{ROADMAP}}$ in Eq. (2), $\tau = 0.01$ and $\rho = 100$ for $\mathcal{L}_{\text{SupAP}}$ in Eq. (5), $\alpha = 0.9$ and $\beta = 0.6$ for $\mathcal{L}_{\text{calibr.}}$ in Eq. (7). We study more in depth the impact of those parameters in Section 4.3. Deep models are trained using Adam [17] for ResNet-50 backbones and AdamW [19] for DeiT transformers [34].
Test protocol Methods are evaluated using the standard recall at k (R@k) and mean average precision at R [24] (mAP@R) metrics (see supplementary Sec. B).

4.1 ROADMAP validation

In this section, all models are trained in the same setting for a fair comparison: we use a ResNet-50 backbone with an embedding size of 512 and a batch size of 64.

Comparison to AP approximations. In Table 1, we compare ROADMAP on the three datasets to recent AP loss approximations including the soft-binning approaches FastAP [3] and SoftBinAP [27], the generic solver BlackBox [28], and the smooth rank approximation [2]. We use the publicly available PyTorch implementations of all these baselines. We can see that ROADMAP outperforms all the current AP approximations by a large margin. The gain is especially pronounced on the large

scale dataset INaturalist. This highlights the importance our two contributions, *i.e.* our robust smooth AP upper bound and our AP decomposability improvement (see supplementary Sec. B).

Table 1: Comparison between ROADMAP and state-of-the-art AP ranking based methods.

	CUB		SOP		INaturalist	
Method	R@1	mAP@R	R@1	mAP@R	R@1	mAP@R
FastAP [3]	58.9	22.9	78.2	51.3	53.5	19.6
SoftBin [27]	61.2	24.0	80.1	53.5	56.6	20.1
BlackBox [28]	62.6	23.9	80.0	53.1	52.3	15.2
SmoothAP [2]	62.1	23.9	80.9	54.6	59.8	20.7
ROADMAP	**64.2**	**25.3**	**82.0**	**56.5**	**64.5**	**25.1**

Comparison to memory methods. We experimented the Cross batch memory (XBM) [38] GitHub code, reaching 80.6 on SOP *vs* 82 for ROADMAP in Table 1 (see supplementary Sec. B). In addition, XBM stores the embeddings of the whole dataset (60k images for SOP), resulting in a training time overhead of $\sim$x3 compared to ROADMAP. This becomes critical on INaturalist, where training while storing 60k images takes about 3 days, and reaches only a R@1 of 60. This shows the suitability of ROADMAP on large-scale settings.

Ablation study. To study more in depth the impact of our contributions, we perform ablation studies in Table 2. We show the improvement against SmoothAP [2] when changing the sigmoid by H^+ and H^- for $\mathcal{L}_{\text{SupAP}}$ in Eq. (5), and the use of $\mathcal{L}_{\text{calibr.}}$ in Eq. (7). We can see that $\mathcal{L}_{\text{SupAP}}$ consistently improves performances over $\mathcal{L}_{\text{SmoothAP}}$ (0.9pt on CUB, 0.5pt on SOP and 1.5pt on INaturalist). $\mathcal{L}_{\text{SupAP}}$ and $\mathcal{L}_{\text{calibr.}}$ equally contribute to the overall gain in CUB and SOP, but the gain of $\mathcal{L}_{\text{calibr.}}$ is much more important on INaturalist. This is explained by the fact that the batch vs. dataset ratio size $\frac{B}{N}$ is tiny ($\ll 1$), making the decomposability gap in Eq. (6) huge. We can see that $\mathcal{L}_{\text{calibr.}}$ is very effective for reducing this gap and brings a gain of more than 3pt.

Table 2: Ablation study for the impact of our two contribution on and the SmoothAP baseline.

			CUB		SOP		INaturalist	
Method	H^-	$\mathcal{L}_{\text{calibr.}}$	R@1	mAP@R	R@1	mAP@R	R@1	mAP@R
SmoothAP [2]	✗	✗	62.1	23.9	80.9	54.6	59.7	20.7
SupAP	✓	✗	62.9	24.6	81.4	55.3	61.2	21.3
ROADMAP	✓	✓	**64.2**	**25.3**	**82.0**	**56.5**	**64.5**	**25.1**

4.2 State of the art comparison

We compare ROADMAP to other state of the art methods across three image retrieval datasets and report the results in Table 3. We divide competitor methods into three categories: metric learning [29, 37, 44, 16, 38, 41], classification losses for image retrieval [45, 43, 1, 33], and AP approximations [3, 28, 2]. ROADMAP falls in the latter category. We use the same setup as in Section 4.1 and follow standard practices for ResNet-50 [33, 41, 1] by using larger images (256×256 on SOP and CUB) and using max instead of average pooling and layer normalization for CUB.

Using the popular ResNet-50 backbone, ROADMAP establishes a new state of the art across all methods for SOP and the challenging INaturalist dataset and outperforms all previous AP approximations on CUB, while being competitive with the other two top performers (ProxyNCA++ and SEC). R@k improvements are consistent on all datasets with a $\sim$2pts R@1 increase on INaturalist and $\sim$3pts increase on SOP compared to SmoothAP, the best performing AP approximation from the literature.

Switching the backbone to the more recent vision transformer architecture DeiT [5, 34], further lifts the performances of ROADMAP by several point, from 3 to 9 points depending on the dataset, with a smaller embedding size (384 *vs* 512). The decomposable AP approximation ROADMAP also outperforms by a significant margin IRT_R, the DeiT architecture for image retrieval introduced in [7] trained with a contrastive loss. Overall ROADMAP achieves state-of-the-art performances across all three datasets by a significant margin.

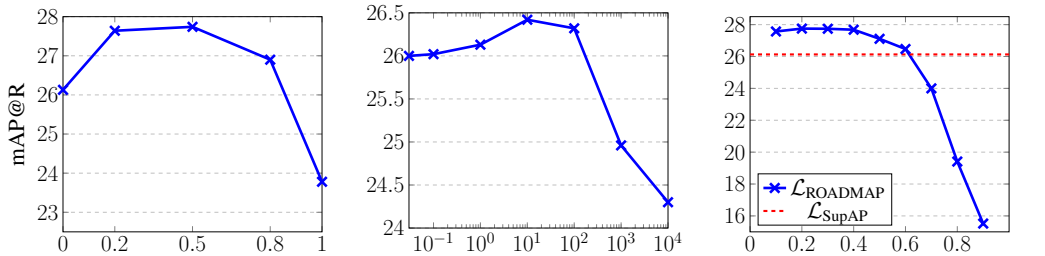
4.3 Model Analysis

We show in Fig. 4 the impact of the main ROADMAP hyperparameters on INaturalist. The relative weighting λ from Eq. (2) controls the balance between our two training objectives $\mathcal{L}_{\text{SupAP}}$ and $\mathcal{L}_{\text{calibr.}}$:

Table 3: Comparison of state of the art performances from the literature on SOP, CUB and INaturalist with the proposed ROADMAP (recall@k). Except for the DeiT category, all methods rely on a standard convolutional backbone (generally ResNet-50).

	Method	dim	SOP			CUB				INaturalist			
			1	10	100	1	2	4	8	1	4	16	32
Metric learning	Triplet SH [39]	512	72.7	86.2	93.8	63.6	74.4	83.1	90.0	58.1	75.5	86.8	90.7
	LiftedStruct [31]	512	62.1	79.8	91.3	47.2	58.9	70.2	80.2	-	-	-	-
	MIC [29]	512	77.2	89.4	95.6	66.1	76.8	85.6	-	-	-	-	-
	MS [37]	512	78.2	90.5	96.0	65.7	77.0	86.3	91.2	-	-	-	-
	SEC [44]	512	78.7	90.8	96.6	68.8	79.4	87.2	92.5	-	-	-	-
	HORDE [16]	512	80.1	91.3	96.2	66.8	77.4	85.1	91.0	-	-	-	-
	XBM [38]	128	80.6	91.6	96.2	65.8	75.9	84.0	89.9	-	-	-	-
	Triplet SCT [41]	512/64	81.9	92.6	96.8	57.7	69.8	79.6	87.0	-	-	-	-
Classification	ProxyNCA [23]	512	73.7	-	-	49.2	61.9	67.9	72.4	61.6	77.4	87.0	90.6
	ProxyGML [45]	512	78.0	90.6	96.2	66.6	77.6	86.4	-	-	-	-	-
	NSoftmax [43]	512	78.2	90.6	96.2	61.3	73.9	83.5	90.0	-	-	-	-
	NSoftmax [43]	2048	79.5	91.5	96.7	65.3	76.7	85.4	91.8	-	-	-	-
	Cross-Entropy [1]	2048	81.1	91.7	96.3	69.2	79.2	86.9	91.6	-	-	-	-
	ProxyNCA++ [33]	512	80.7	92.0	96.7	69.0	79.8	87.3	92.7	-	-	-	-
	ProxyNCA++ [33]	2048	81.4	92.4	96.9	72.2	82.0	89.2	93.5	-	-	-	-
AP loss	FastAP [3]	512	76.4	89.0	95.1	-	-	-	-	60.6	77.0	87.2	90.6
	BlackBox [28]	512	78.6	90.5	96.0	64.0	75.3	84.1	90.6	62.9	79.4	88.7	91.7
	SmoothAP [2]	512	80.1	91.5	96.6	-	-	-	-	67.2	81.8	90.3	93.1
	SoftBin* [27]	512	80.6	91.3	96.1	61.2	73.14	83.0	89.5	64.2	77.1	82.7	91.7
	ROADMAP (ours)	512	83.1	92.7	96.3	68.5	78.7	86.6	91.9	69.1	83.1	91.3	93.9
DeiT	IRT_R [7]	384	84.2	93.7	97.3	76.6	85.0	91.1	94.3	-	-	-	-
	ROADMAP (ours)	384	**86.0**	**94.4**	**97.6**	**77.4**	**85.5**	**91.4**	**95.0**	**73.6**	**86.2**	**93.1**	**95.2**

$\lambda = 0$ reduces $\mathcal{L}_{\text{ROADMAP}}$ to $\mathcal{L}_{\text{SupAP}}$ while $\lambda = 1$ to $\mathcal{L}_{\text{calibr.}}$. We can see in Fig. 4a that training with the complete $\mathcal{L}_{\text{ROADMAP}}$ with both $\mathcal{L}_{\text{calibr.}}$ and $\mathcal{L}_{\text{SupAP}}$ is always better than using only one of the two losses. Note that results are stable in the $[0.2, 0.8]$ range with a consistent $\sim$1.5pt increase, demonstrating the robustness of ROADMAP to this hyperparameter tuning.



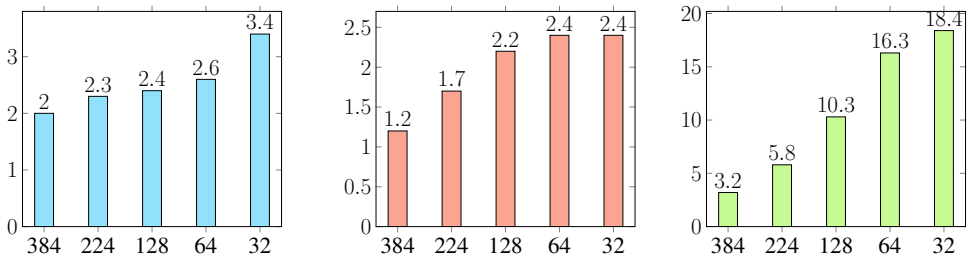
(a) mAP@R *vs* λ for $\mathcal{L}_{\text{ROADMAP}}$ (b) mAP@R *vs* ρ for $\mathcal{L}_{\text{SupAP}}$ (c) mAP@R *vs* $\alpha - \beta$ for $\mathcal{L}_{\text{calibr.}}$

Figure 4: Analysis of ROADMAP hyperparameters on INaturalist (batch size 224).

Fig. 4b shows the influence of the slope ρ that controls the linear regime in H^- and determines the amount of gradient backpropagated for negative samples with a (wrong) high score. As shown in Fig. 4b, the improvement is important and stable in $[10, 100]$. Note that $\rho > 0$ already improves the results compared to $\rho = 0$ in [2]. There is an important decrease when $\rho \gg 100$ probably due to the high gradient that takes over the signal for correctly ranked samples.
The impact of the margin $\alpha - \beta$ in $\mathcal{L}_{\text{calibr.}}$ is shown in Fig. 4c. Once again, ROADMAP exhibits a robust behaviour w.r.t. the values of its hyperparameters: any margin in the $[0.1, 0.6]$ range results in

an improvement in mAP@R compared to the $\mathcal{L}_{\text{SupAP}}$ baseline without the decomposability loss. Best results are achieved with smaller margins $0.1 < \alpha - \beta < 0.4$.

Fig. 5 shows the improvement in mAP@R on the three datasets when adding $\mathcal{L}_{\text{calibr.}}$ to $\mathcal{L}_{\text{SupAP}}$. We can see that the increase becomes larger as the batch size gets smaller. This confirms our intuition that the decomposability in $\mathcal{L}_{\text{calibr.}}$ has a stronger effect on smaller batch sizes, for which the AP estimation is noisier and DG_{AP} larger. This is critical on the large-scale dataset INaturalist where the batch AP on usual batch sizes is a very poor approximation of the global AP.



(a) CUB (b) SOP (c) INaturalist

Figure 5: Relative increase of the mAP@R *vs* batch size when adding $\mathcal{L}_{\text{calibr.}}$ to $\mathcal{L}_{\text{SupAP}}$.

As a qualitative assessment, we show in Fig. 6 some results of ROADMAP on INaturalist. We show the queries (in purple) and the 4 most similar retrieved images (in green). We can appreciate the semantic quality of the retrieval. More qualitative results are provided in supplementary Sec. C.

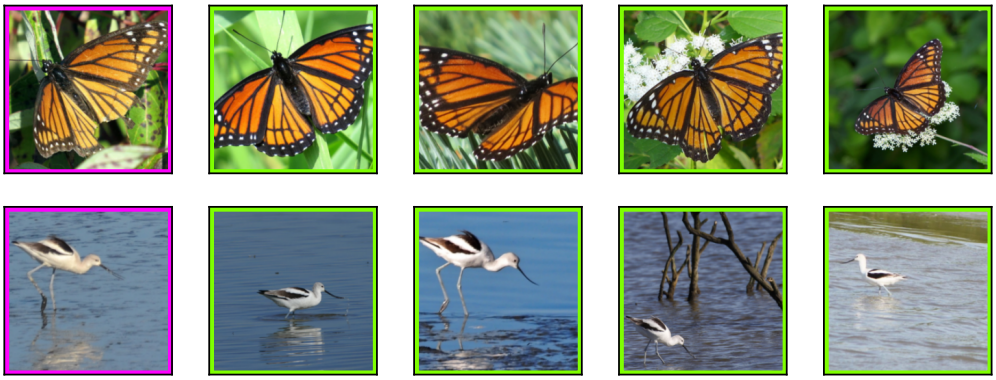


Figure 6: Results on INaturalist: a query (purple) with the 4 most similar retrieved images (green).

5 Conclusion

This paper introduces the ROADMAP method for gradient-based optimization of average precision. ROADMAP is based on a smooth rank approximation, leading to the $\mathcal{L}_{\text{SupAP}}$ being both accurate and robust. To overcome the lack of decomposability in AP, ROADMAP is equipped with a calibration loss $\mathcal{L}_{\text{calibr.}}$ which aims at reducing the decomposability gap. We provide theoretical guarantees as well as experiments to assess this behavior. Experiments show that ROADMAP can combine the strength of ranking methods with the simplicity of a batch strategy. Without bells and whistles, ROADMAP is able to outperform state-of-the-art performances on three datasets, and remains effective even with small batch sizes.

As any work on image retrieval, our contribution could be applied to critical applications in surveillance scenarios, *e.g.* face recognition or person re-identification. ROADMAP is neither worse nor better than previous work in this regard. Our work is also a data-driven learning method, and thus inherits the risk of perpetuating dataset biases. Future work will focus on improving fair and accurate retrieval by reducing dataset biases. We also plan to relax the need for full supervision to tackle situations more representative to in-the-wild scenarios.

References

[1] Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *European Conference on Computer Vision*, pages 548–564. Springer, 2020.

[2] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *European Conference on Computer Vision*, pages 677–694. Springer, 2020.

[3] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1861–1870, 2019.

[4] Micael Carvalho, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz, editors, *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 35–44. ACM, 2018.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Thibaut Durand, Nicolas Thome, and Matthieu Cord. Exploiting negative evidence for deep latent structured models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):337–351, 2019.

[7] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou. Training vision transformers for image retrieval. *arXiv preprint arXiv:2102.05644*, 2021.

[8] Martin Engilberge, Louis Chevallier, Patrick Perez, and Matthieu Cord. Sodeep: A sorting deep net to learn ranking loss surrogates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 12. BMVA Press, 2018.

[10] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[11] Albert Gordo, Jon Almazán, Jérôme Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *Int. J. Comput. Vis.*, 124(2):237–254, 2017.

[12] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[13] Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[14] Kun He, Fatih Cakir, Sarah Adel Bargal, and Stan Sclaroff. Hashing as tie-aware learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] Kun He, Yan Lu, and Stan Sclaroff. Local descriptors optimized for average precision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] Pierre Jacob, David Picard, Aymeric Histace, and Edouard Klein. Metric learning with horde: High-order regularizer for deep embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6539–6548, 2019.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Marc T. Law, Nicolas Thome, and Matthieu Cord. Learning a distance metric from relative comparisons between quadruplets of images. *Int. J. Comput. Vis.*, 121(1):65–94, 2017.

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[20] R. Manmatha, Chao-Yuan Wu, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2859–2867. IEEE Computer Society, 2017.

[21] Brian Mcfee and Gert Lanckriet. Metric learning to rank. In *In Proceedings of the 27th annual International Conference on Machine Learning (ICML*, 2010.

[22] Pritish Mohapatra, Michal Rolínek, C.V. Jawahar, Vladimir Kolmogorov, and M. Pawan Kumar. Efficient optimization for rank-based loss functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[23] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017.

[24] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020.

[25] Marin Vlastelica P., Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. In *ICLR*, 2020.

[26] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, volume 9905 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2016.

[27] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5107–5116, 2019.

[28] Michal Rolínek, Vít Musil, Anselm Paulus, Marin Vlastelica, Claudio Michaelis, and Georg Martius. Optimizing rank-based metrics with blackbox differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7620–7630, 2020.

[29] Karsten Roth, Biagio Brattoli, and Bjorn Ommer. Mic: Mining interclass characteristics for improved metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8000–8009, 2019.

[30] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[31] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[32] Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. Stochastic class-based hard example mining for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[33] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *European Conference on Computer Vision (ECCV)*. Springer, 2020.

[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[35] Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[36] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[37] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019.

[38] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6388–6397, 2020.

[39] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017.

[40] Eric Xing, Michael Jordan, Stuart J Russell, and Andrew Ng. Distance metric learning with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2003.

[41] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful. In *European Conference on Computer Vision*, pages 126–142. Springer, 2020.

[42] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. 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*, SIGIR '07, pages 271–278, New York, NY, USA, 2007. ACM.

[43] Andrew Zhai and Hao-Yu Wu. Making classification competitive for deep metric learning. *CoRR*, abs/1811.12649, 2018.

[44] Dingyi Zhang, Yingming Li, and Zhongfei Zhang. Deep metric learning with spherical embedding. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18772–18783. Curran Associates, Inc., 2020.

[45] Yuehua Zhu, Muli Yang, Cheng Deng, and Wei Liu. Fewer is more: A deep graph metric learning perspective using fewer proxies. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17792–17803. Curran Associates, Inc., 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 3 for details of the two contributions.
   (b) Did you describe the limitations of your work? [Yes] See Section 5 for the discussion of the limitations of our work.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] Although our work is more computationally efficient than other recent approaches, our work a data-driven learning method prone to data biases and discriminations, which can have a negative impact for applications such as surveillance or re-identification systems. We further discuss this point in Section 5.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3 for theoretical results ($\mathcal{L}_{\text{SupAP}}$ as an upper bound of AP and improved bound using $\mathcal{L}_{\text{calibr.}}$).

(b) Did you include complete proofs of all theoretical results? [Yes] The paper contains derivation of the proof, and steps are detailed in supplementary material.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Giving access to the code has yet to be authorized internally. However, we gave the main hyperparameters of our method in the paper. In the supplementary we further detail the training procedure to run our experiments and reproduce our results. We plan to release the code in the future.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 for data splits and main hyperparameters and supplementary material for details.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Due to the intensive computations required, we do not report results on repeated runs for every experiments. However, we report results with thorough ablations and validations in various setting (multiple datasets and hyperparameters combinations) that demonstrate statistical significance of the improvements. Our experiments also follow standard practices of the community when reporting state of the art performances. Moreover, we include error bars in the supplementary section B for the comparison to the AP baselines.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We discussed the computational cost of our approach compared to recent works in the main submission. We detailed in the supplementary the training resources needed and the type of hardware used.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4 for datasets and baseline references and we provide the link of all implementations used in the supplementary.

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] INaturalist and CUB are images of animals. SOP are images of products from eBay.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Used datasets contain neither personally identifiable information nor offensive content to the best of our knowledge.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No crowdsourcing or human subjects.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]