Attention-Based Query Expansion Learning

Albert Gordo[0000−0001−9229−4269], Filip Radenovic[0000−0002−7122−2765], and Tamara Berg[0000−0002−1272−3359]

Facebook AI

Abstract. Query expansion is a technique widely used in image search consisting in combining highly ranked images from an original query into an expanded query that is then reissued, generally leading to increased recall and precision. An important aspect of query expansion is choosing an appropriate way to combine the images into a new query. Interestingly, despite the undeniable empirical success of query expansion, ad-hoc methods with different caveats have dominated the landscape, and not a lot of research has been done on *learning* how to do query expansion. In this paper we propose a more principled framework to query expansion, where one trains, *in a discriminative manner*, a model that learns how images should be aggregated to form the expanded query. Within this framework, we propose a model that leverages a self-attention mechanism to effectively learn how to transfer information between the different images before aggregating them. Our approach obtains higher accuracy than existing approaches on standard benchmarks. More importantly, our approach is the only one that consistently shows high accuracy under different regimes, overcoming caveats of existing methods.

Keywords: image retrieval, query expansion learning, attention-based aggregation

1 Introduction

Image search is a fundamental task in computer vision, directly applied in a number of applications such as visual place localization [21,39,2], 3D reconstruction [16,40,24], content-based image browsing [50,27,1], *etc.* Image search is typically cast as a nearest neighbor search problem in the image representation space, originally using local feature matching and bag-of-words-like representations [43], and, more recently, CNN-based global image representations [13,33].

To increase the accuracy of image search systems, a robust representation of the query image is desirable. Query expansion (QE) is a commonly used technique to achieve this goal, where relevant candidates produced during an initial ranking are aggregated into an expanded query, which is then used to search more images in the database. Aggregating the candidates reinforces the information shared between them and injects new information not available in the original query. This idea was originally exploited in the work of Chum *et al.* [7], introducing the first attempt at image retrieval QE. This averaging of query and top ranked

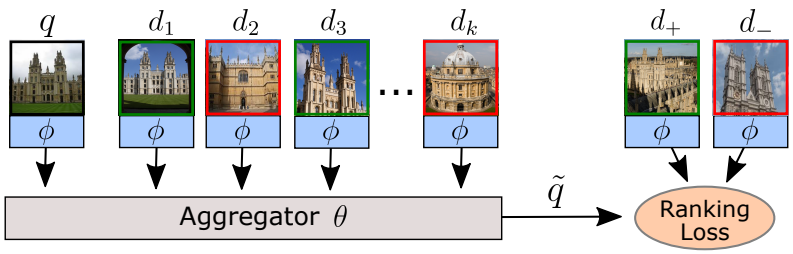

Fig. 1. Outline of our proposed approach. During training, we sample a query q and its nearest neighbors in the training dataset (where their features have been precomputed with the function ϕ, typically a CNN) and use our proposed attention-based model θ to aggregate them into an expanded query $\tilde{\mathbf{q}}$. Given positive ($\mathbf{d}_+$) and/or negative ($\mathbf{d}_-$) samples, we use a ranking loss to optimize θ. Images with the **green** (**red**) border represent relevant (non-relevant) samples to the query. At inference, we construct the expanded $\tilde{\mathbf{q}}$ given $\mathbf{q}$ and its neighbors in the index, and use it to query the index again.

results [7], or ad-hoc variations of it [6,45,3,13,33], are now used as a standard method of performance boosting in image retrieval.

Selecting which images from the initial ranking should be used in the QE procedure is however a challenging problem, since we do not have guarantees that they are actually relevant to the query. Early methods use strong geometrical verification of local features to select true positives [7,6,3,45]. As CNN-based global features lack this possibility, the most common approach is to use the k-nearest neighbors to the query [13,33], potentially including false positives. Yet, if k is larger than the number of relevant images, topic drift will degrade the results significantly. This leads to two unsatisfying alternatives: either use a very small k, potentially not leveraging relevant images, or use weighted average approaches with decreasing weights as a function of ranking [13] or image similarity [33], where setting the appropriate decay is a task just as challenging as choosing the optimal k. This has unfortunately led to many works tuning the k parameter directly on test, as well as to use different values of k for each dataset. Replacing k-nearest neighborhoods with similarity-based neighborhoods turn out to be just as unstable, as, unlike inlier count for local features, cosine similarity of CNN global features is not directly comparable between different query images [29].

We argue that existing QE approaches are generally not robust and use ad-hoc aggregation methods, and instead propose to cast QE as a discriminative learning problem. Similar to recent methods that learn embeddings suitable for image retrieval using large-scale datasets [13,33], we formulate the problem as a ranking one, where we train an aggregator that produces the expanded query, optimized to rank relevant samples ahead of non-relevant ones, *cf.* Figure 1. We use a large-scale dataset, disjoint from the evaluation ones, to train and validate our model and its parameters. We then leverage a self-attention mechanism to design an aggregator model that can transfer information between the candidates (Figure 2), enabling the model to learn the importance of each sample before aggregating them. We call this model Learnable Attention-based Query Expansion, or LAttQE. Unlike

previous QE approaches, LAttQE does not produce monotonically decreasing weights, allowing it to better leverage the candidates in the expansion. LAttQE is more robust to the choice of k thanks to the large-scale training, which enables the model to better handle false positive amongst the top neighbors, and is usable across a wide range of class distributions without sacrificing the performance at any number of relevant images.

Our contributions are as follows: (i) We show that standard query expansion methods, albeit seemingly different, can be cast under the same mathematical framework, allowing one to compare their advantages and shortcomings in a principled way. (ii) We propose to treat query expansion as a discriminative learning problem, where an aggregation model is learned in a supervised manner. (iii) We propose LAttQE, an aggregation model designed to share information between the query and the top ranked items by means of self-attention. We extend this query expansion model to also be useful for database-side augmentation. (iv) We show that our proposed approach outperforms commonly-used query expansion methods in terms of both accuracy and robustness on standard benchmarks.

2 Related work

Image retrieval query expansion. Average query expansion (AQE) in image retrieval was originally proposed for representations based on local features [7], and tuned for the bag-of-words search model [43], where local features are aggregated after a strict filtering step, usually based on strong feature geometry [7,6] or Hamming embedding distance [45]. For CNN-based global image representation, AQE is implemented by mean-aggregating the top k retrieved images [13,33]. It has been argued that setting an optimal k for several datasets of different positive image distributions is a non-trivial task [33]. Instead, Gordo *et al.* [13] propose using a weighted average, where the weight is a monotonically decaying function over the rank of retrieved images. We denote this method as average query expansion with decay, or AQEwD. Likewise, Radenovic *et al.* [33] use a weighted average, where the weights are computed as a power-normalized similarity between the query and the top ranked images. This method, known as alpha query expansion (αQE), has proven to be fairly robust to the number of neighbors k, and is used as a *de facto* standard by a number of recent state-of-the-art image retrieval works [34,36,14,18,11,17]. Finally, Arandjelovic *et al.* [3] proposed discriminative query expansion (DQE) where they train a linear SVM using top ranked images as positives, and low ranking images as negatives, and use the resulting classifier as the expanded query. Note that this is very different from our method, as DQE trains independent classifiers for each query, while we train one single model using a large disjoint dataset.

Image retrieval database pre-processing. If the database is fixed at indexing time, one can pre-process the database to refine the image representations and improve the accuracy accuracy. Database-side augmentation (DBA) [3] is a method that applies QE to each image of the database and replaces the original representation of the image by its expanded version. Although it increases the

offline pre-processing time, it does not increase the memory requirements of the pipeline or the online search time. All aggregation-based QE methods described in the previous paragraph [7,13,3,33] can be applied as different flavors of DBA, including our proposed LAttQE. A different line of work [32,42,8] indexes local neighborhoods of database images together with their respective representations, in order to refine the search results based on the reciprocal neighborhood relations between the query and database images. Besides offline pre-processing, these approaches require additional storage and are slower at query time. Finally, some works [19,5] build a nearest neighbor graph using the database image representations and traverse it at query time, or, alternatively, encode graph information into image descriptors [23]. It increases the amount of required memory by storing the graph structure of the database, and increases online search complexity by orders of magnitude. Both reciprocal-nearest-neighbor and graph-based methods are complementary to our work, and can be applied after augmenting the database representations with our method. When dealing with dynamically-growing indexes, applying these methods becomes even more challenging, which makes them generally unappealing despite the accuracy gains.

Self-attention. The self-attention transformer [47] has established itself as the core component of strong language representation models such as BERT [10] or GPT-2 [35] due to its ability to capture complex interactions between tokens and due to how easy it is to increase the capacity of models simply by stacking more encoders. Self-attention has also shown applications outside of NLP. Wang *et al.* [48] leverage self-attention to aggregate descriptors from different parts of the image in order to capture interactions between them in a non-local manner. In a similar way, Girdhar and Ramanan [12] use self-attention as an approximation for second order pooling. In a different context, Lee *et al.* [22] use self-attention as a graph pooling mechanism to combine both node features and graph topology in the pooling. In this paper we use self-attention as a way to transfer information between the top k results so we can construct a more discriminative query. As we describe in Section 3, self-attention is an excellent mechanism to this end.

Query expansion and relevance feedback in information retrieval. The information retrieval community has leveraged query expansion techniques for several decades [26,37,4]. Most interestingly, in the information retrieval community, query expansion methods expand or reformulate query terms independently of the query and results returned from it, via, *e.g.*, reformulation with a thesaurus [25]. What the image search community denotes as query expansion is generally known as relevance feedback (RF), and more precisely, pseudo-RF, as one generally does not have access to the true relevance of the neighbors – although a case could be made for geometrical verification methods [7] providing explicit feedback. Our focus in this work is not on information retrieval methods for two reasons: (i) they generally deal with explicit or implicit RF instead of pseudo-RF; (ii) they generally assume high-dimensional, sparse features (*e.g.* bags of terms), and learn some form of term weighting that is not applicable in our case.

3 Attention-based query expansion learning

We start this section by presenting a generalized form of query expansion, and by showing that well-known query expansion methods can be cast under this framework. We then propose a general framework for learning query expansion in a discriminative manner. Last, we propose LAttQE (Learnable Attention-Based Query Expansion), an aggregation model that leverages self attention to construct the augmented query and that can be trained within this framework.

3.1 Generalized query expansion

We assume that there exists a known function $\phi : \Omega \to \mathcal{R}^D$ that can embed items (*e.g.* images) into an l_2-normalized D-dimensional vectorial space. For example, ϕ could be a CNN trained to perform image embedding [13,33,36]. Let us denote with q a query item, and, following standard convention of using bold typeface for vectors, let us denote with $\mathbf{q} = \phi(q)$ its D-dimensional embedding. Similarily, let us denote with $\{\mathbf{d}\}^k = \mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_k$ the embeddings of the top k nearest neighbors of $\mathbf{q}$ in a dataset $\mathcal{D}$ according to some measure of similarity, *e.g.* the cosine similarity, and sorted in decreasing order. Let us also denote with $\{\mathbf{d}\}^-$ a collection of dataset items that are not close to the query, according to the same measure of similarity. Last, for convenience, let us alias $\mathbf{d}_0 := \mathbf{q}$.

We propose the following generalized form of query expansion:

$$\hat{\mathbf{q}} = \frac{1}{Z} \sum_{i=0}^{k} \theta(\mathbf{d}_i \mid \mathbf{q},\ \{\mathbf{d}\}^k,\ \{\mathbf{d}\}^-,\ i), \tag{1}$$

where Z is a normalization factor, and θ is a learnable function that takes an individual sample and applies a transformation conditioned on the original query $\mathbf{q}$, the top k retrieved results $\{\mathbf{d}\}^k$, a collection of low-ranked samples $\{\mathbf{d}\}^-$, and its position i in the ranking. The final augmented query is computed by aggregating the transformed top k results, including the query, and applying a normalization Z (*e.g.* ℓ_2 normalization)[1].

Standard query expansion methods can be cast under this framework. In fact, they can be cast under a more constrained form: $\theta(\mathbf{d}_i \mid \mathbf{q},\ \{\mathbf{d}\}^k,\ \{\mathbf{d}\}^-,\ i) = w_i \mathbf{d}_i$, where the value of w_i is method-dependent, see Table 1. Two things are worth noticing. First, for all methods, w_i depends either on positional information (*e.g.* the sample got ranked at position i out of k, as done by AQEwD), or on information about the content (*e.g.* the power-normalized similarity between the item and the query, as done by αQE). None of the methods leverage both the positional and the content information simultaneously. Second, except for DQE, all methods produce a monotonically decreasing $\mathbf{w}$, *i.e.*, if $i > j$, then $w_i \leq w_j$. The implication is that these methods do not have the capacity to uplift the samples amongst the top k retrieved results that are indeed relevant to the query but were ranked after some non-relevant samples. That is, any top-ranked,

[1] Note that Eqn. (1) does not aggregate over $\{\mathbf{d}\}^-$. This is just to ease the exposition; negative samples can also be aggregated if the specific method requires it, *e.g.*, DQE.

Method	$\theta(\mathbf{d}_i \mid \mathbf{q}, \{\mathbf{d}\}^k, \{\mathbf{d}\}^-, i) = w_i \mathbf{d}_i$
[7] **AQE:** Average QE	$w_i = 1$
[13] **AQEwD:** AQE with decay	$w_i = (k - i)/k$
[3] **DQE:** Discriminative QE	$\mathbf{w}$ is the dual-form solution of an SVM optimization problem using $\{\mathbf{d}\}^k$ as positives and $\{\mathbf{d}\}^-$ as negatives
[33] **αQE:** α-weighted QE	$w_i = \text{sim}(\mathbf{q}, \mathbf{d}_i)^\alpha$, with α being a hyperparameter.

Table 1. Standard query expansion (QE) methods and their associated transformations. More details about the methods can be found in Section 2.

non-relevant item will contribute more to the construction of the expanded query than any relevant item ranked after it, with clear negative consequences.

3.2 Query expansion learning

We propose that, following recent approaches in representation learning [33,13], one can learn a differentiable θ transformation in a data-driven way (Figure 1). This training is done in a supervised manner, and ensures that relevant items to the (expanded) query are closer to it than elements that are not relevant. This is achieved by means of losses such as the triplet loss [49] or the contrastive loss [15]. The approach requires access to an annotated dataset (*e.g.* rSfM120k [33]), but the training data and classes used to learn θ can be disjoint from the pool of index images that will be used during deployment, as long as the distributions are similar. From that point of view, the requirements are similar to other existing image embedding learning methods in the literature.

At training time, besides sampling queries, positive, and negative samples, one also has to consider the nearest neighbors of the query for the expansion. Sampling a different subset of neighbors each time, as a form of data augmentation, can be useful to improve the model robustness. We provide more details about the process in the experimental section. Finally, we note that this framework allows one to learn θ and ϕ jointly, as well as to learn how to perform QE and DBA jointly, but we consider those variations out of the scope of this work.

3.3 Learnable Attention-based Query Expansion (LAttQE)

We propose a more principled θ function that overcomes the caveats of previous methods, and that can be trained using the framework described in the previous section. In particular, our θ function is designed to be capable of transferring information between the different retrieved items, giving all top-ranked relevant samples the opportunity to significantly contribute to the construction of the expanded query. To achieve this we rely on a self-attention mechanism. We leverage the transformer-encoder module developed by Vaswani *et al.* [47], where, in a nutshell, a collection of inputs first share information through a multi-head attention mechanism, and later are reprojected into an embedding space using

fully connected layers with layer normalization and residual connections – see Fig. 1 of Vaswani *et al.* [47] for a diagram of this module (left) and the decoder module (right), not used in this work. Stacking several of these encoders increases the capacity of the model and enables sharing more contextual information. The exact mechanism that the stack of self-attention encoders uses to transfer information is particularly suited for our problem:

1. The encoder's scaled dot-product attention [47] performs a weighted sum of the form $\sum_{j=0}^{k} \text{Softmax}(\mathbf{d}_i^T [\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_k] / C)_j \mathbf{d}_j$, where C is a constant, in practice computing the similarity between $\mathbf{d}_i$ and all other inputs and using that as weights to aggregate all the inputs. Observing equations (1) and (3), one can see self-attention as a way to perform expansion of the input samples, leading to richer representations that are then used to compute the weights.

2. The multihead attention enables focusing on different parts of the representations. This is important because computing similarities using only the original embedding will make it difficult to change the original ranking. By using multihead attention, we discover parts of the embeddings that are still similar between relevant items and dissimilar between non-relevant items, permitting the model to further upweight relevant items and downweight non-relevant ones.

3. Under this interpretation of the encoder, the stack of encoders allows the model to "refine" the expansion process in an iterative manner. One can see this as expanding the queries, making a first search, using the new neighbors to expand a better query, find new neighbors, etc. Although the pool of neighbors remains constant, we expect the expansion to become more and more accurate.

Aggregation. The stack of encoders takes the query $\mathbf{q}$ and the top results $\mathbf{d}_1 \ldots \mathbf{d}_k$ as input, and produces outputs $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{d}}_1 \ldots \tilde{\mathbf{d}}_k$. To construct the expanded query, a direct solution consists in aggregating them (*e.g.* through average or weighted average) into a single vector that represents the expanded query. However, this is challenging in practice, as it requires the encoder to learn how to create outputs that lie in the same space as the original data, something particularly hard when the embedding function ϕ is not being simultaneously learned. We empirically verify that learning such a function leads to weak results. Although we speculate that learning a "direct" θ function jointly with ϕ could lead to superior results, the practical difficulties involved in doing so make this approach unappealing. Instead, to ensure that we stay in a similar space, we relax the problem and also construct the expanded query as a weighted sum of the top k results, where the weights $\mathbf{w}$ are predicted by our model. If we denote with M the stack of encoders, the transformed outputs can be represented as

$$\tilde{\mathbf{d}}_i = M(\{\mathbf{q}\} \cup \{\mathbf{d}\}^k)_i. \tag{2}$$

Then, inspired by other methods such as αQE, we can construct the weight w_i as the similarity between item $\mathbf{d}_i$ and the query $\mathbf{q}$ *in the transformed space*, *i.e.*, $w_i = \text{sim}(\tilde{\mathbf{q}}, \tilde{\mathbf{d}}_i)$. This leads to our proposed θ:

$$\theta(\mathbf{d}_i \mid \mathbf{q},\ \{\mathbf{d}\}^k,\ \{\mathbf{d}\}^{\text{-}},\ i) = \text{sim}(\tilde{\mathbf{q}}, \tilde{\mathbf{d}}_i)\mathbf{d}_i. \tag{3}$$

Including rank information. As presented, the proposed method does not leverage in any way the ranking of the results. Indeed, the encoders see the inputs

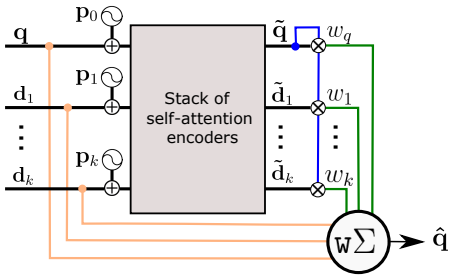

Fig. 2. Proposed aggregator. The output $\hat{\mathbf{q}}$ is constructed as the weighted sum ($\mathtt{w}\Sigma$) of the query $\mathbf{q}$ and the nearest neighbors $\mathbf{d}_1 \dots \mathbf{d}_k$. The weights are computed by running the inputs through a stack of self-attention encoders after including positional information ($\circlearrowleft$) and computing the similarity (through a normalized dot product $\otimes$) between the transformed query $\tilde{\mathbf{q}}$ and all the transformed samples $\tilde{\mathbf{d}}_1 \dots \tilde{\mathbf{d}}_k$.

as a set, and not as a sequence of results. This prevents the model from leveraging this information, *e.g.* by learning useful biases such as "top results tend to be correct, so pay more attention to them to learn the transformations". To enable the model to reason not only about the content of the results but also about their ranking, we follow standard practice when dealing with transformers and include a positional encoding that is added to the inputs before being consumed by the encoder, *i.e.*, $\text{pe}(\mathbf{d}_i) = \mathbf{d}_i + \mathbf{p}_i$, where each $\mathbf{p}_i \in \mathcal{R}^D$ is a learnable variable within our model. The full proposed aggregator that leverages θ with positional encoding is depicted in Figure 2.

Auxiliary classification loss. Since, at training time, we have access to the annotations of the images, we know which of the top k results are relevant to the query and which ones are not. This enables us to have an auxiliary linear classifier that predicts whether $\tilde{\mathbf{d}}_i$ is relevant to the query or not. The role of this classifier, which is only used at train time and discarded at inference time, is to encourage the relevant and non-relevant outputs of the encoder to be linearly separable, inducing the relevant items to be more similar to the query than the non-relevant ones. Our empirical evaluation in Section 4 shows that the use of this auxiliary loss can noticeably increase the accuracy of the model.

3.4 Database-side augmentation

Database-side augmentation (DBA) is technique complementary to query expansion. Although different variations have been proposed [46,3,44,13], the main idea is that one can perform query expansion, *offline*, on the database images. This produces an expanded version of the database images, which are then indexed, instead of indexing the original ones. When issuing a new query, one searches on the expanded index, and not on the original one.

Our proposed approach can also be used to perform better database-side augmentation, using θ to aggregate the top k neighbors of each database image. However, this approach did not work in practice. We believe that the reason is that, on the database side, many images are actually distractors, unrelated to any query, and our model was assigning weights too high for unrelated images when using them as queries. To address this, we propose to use a tempered softmax over the weights, *i.e.*, instead of computing our weights as $w_i = \text{sim}(\tilde{\mathbf{q}}, \tilde{\mathbf{d}}_i)$, we

compute it as

$$w_i = \text{Softmax}(\text{sim}(\tilde{\mathbf{q}}, [\tilde{\mathbf{d}}_0, \tilde{\mathbf{d}}_1, \ldots, \tilde{\mathbf{d}}_k])/T)_i, \quad (4)$$

where $\text{sim}(\tilde{\mathbf{q}}, [\tilde{\mathbf{d}}_0, \tilde{\mathbf{d}}_1, \ldots, \tilde{\mathbf{d}}_k])$ is a vector of similarities between $\tilde{\mathbf{q}}$ and all the $\tilde{\mathbf{d}}$s, and T is a learnable scalar.

To achieve the best results, we employ a curriculum learning strategy, where first we train our model without softmax, and then we freeze the parameters of the model, incorporate the tempered softmax, and continue training while updating only T. This strategy led to a DBA that not only gave the best results in terms of accuracy but that was also more stable than other variants.

4 Experiments

In this section we discuss implementation details of our training, evaluate different components of our method, and compare to the state of the art.

4.1 Training setup and implementation details

Image representation. For all experiments we use a publicly-available, state-of-the-art model for image retrieval [33][2] to extract the underlying features. We use the best-performing model from the project page (trained on Google Landmarks 2018 data [29]), consisting of a ResNet101 trunk followed by generalized-mean pooling and a whitening layer, which produces features of 2048 dimensions. Following [33], we extract features at 3 scales $(1, \sqrt{2}, 1/\sqrt{2})$, mean-aggregate them, and finally ℓ_2-normalize to form the final 2048D representation.

Training dataset. We use the publicly available rSfM120k created by Radenovic *et al.* [33], which comprises images selected from 3D reconstructions of landmarks and urban scenes. These reconstructions are obtained from an unordered image collection using a combined local-feature-based image retrieval and structure-from-motion pipeline. The 3D reconstruction cluster ids serve as a supervision for selecting positive and negative pairs. In total, 91642 images from 551 classes are used for training, while additional 6403 database images – 1691 of which are used as queries – from 162 classes, disjoint from the training ones, are set aside for validation. Performance on validation is measured as mean average precision (mAP) [30] over all 1691 queries.

Learning configuration. To train LAttQE we follow [33] and use a contrastive loss of the form $yz^2 + (1-y)max(0, m-z)^2$, with m being the margin, $z = ||\hat{q} - d||$, and $y \in \{0, 1\}$ denotes whether d is relevant to q or not. We backpropagate through $\hat{q}$, which in turn optimizes the transformers (see Fig 2). Other recent ranking losses [9,36,28] could also be used. Since the base representations are already strong, we use a margin of 0.1, which ensures that positives are pulled together while only pushing away negatives that are too close to the query. LAttQE consists of a stack of 3 transformer encoders, each one with 64 heads. We did not see any improvement after further increasing the capacity of the

[2] github.com/filipradenovic/cnnimageretrieval-pytorch

model. The self-attention and fully-connected layers within the encoders preserve the original dimensionality of the inputs, 2048D. We also follow [33] regarding the sampling strategy for positives and negatives: we select 5 negatives per positive, found in a pool of 20000 samples that gets refreshed every 2000 updates. When sampling neighbors to construct the augmented query, as a form of data augmentation, the exact number of neighbors is drawn randomly between 32 and 64, and neighbors are also randomly dropped according to a Bernoulli distribution (where the probability of dropping neighbors in each query is itself drawn from a uniform distribution between 0 and 0.6). The auxiliary classification head uses a binary cross-entropy loss. We use Adam to optimize the model, with a batch size of 64 samples, a weight decay of $1e$-6, and an initial learning rate of $1e$-4 with an exponential decay of 0.99. The optimal number of epochs (typically between 50 and 100) is decided based on the accuracy on the validation set, and is typically within 1% of the optimal iteration if it was validated directly on test.

4.2 Test datasets and evaluation protocol

Revisited Oxford and Paris. Popular Oxford Buildings [30] and Paris [31] datasets have been revisited by Radenovic *et al.* [34], correcting and improving the annotation, adding new more difficult queries, and updating the evaluation protocol. Revisited Oxford ($\mathcal{R}$Oxford) and Revisited Paris ($\mathcal{R}$Paris) datasets contain 4,993 and 6,322 images respectively, with 70 held out images with regions of interest that are used as queries. Unlike the original datasets, where the full-size version of query images are present in the database side, this is not the case in revisited versions, making query expansion a more challenging task. For each query, the relevant database images were labeled according to the "difficulty" of the match. The labels are then used to define three evaluation protocols for $\mathcal{R}$Oxford and $\mathcal{R}$Paris: Easy (E), Medium (M), and Hard (H). As suggested by Radenovic *et al.* [34], which points out that the Easy protocol is saturated, we only report results on the Medium and Hard protocols. Note that Oxford and Paris landmarks are not present in rSfM120k training and validation datasets.
Distractors. A set of 1 million hard distractor images ($\mathcal{R}$1M) were collected in [34]. These distractors can, optionally, be added to both $\mathcal{R}$Oxford and $\mathcal{R}$Paris to evaluate performance on a more realistic large-scale setup.
We do not evaluate on INRIA Holidays [20], another common retrieval dataset, since performing query expansion on Holidays is not a standard practice.

4.3 Model study

Table 2 displays the results of our proposed model, using all components (row ii), and compares it with the results without query expansion (row i). We use 64 neighbors for query expansion, as validated on the validation set of rSfM120k. Our model clearly improves results on $\mathcal{R}$Oxford and $\mathcal{R}$Paris, both on the M and H settings. We further study the impact of the components introduced in Sec. 3.

		$\mathcal{R}$Oxford		$\mathcal{R}$Paris		
		M	H	M	H	Mean
(i)	No QE	67.3	44.3	80.6	61.5	63.4
(ii)	**Full model**	73.4	49.6	86.3	70.6	70.0
(iii)	Without self-attention	66.0	41.5	86.1	70.2	66.0
(iv)	Without positional encoding	58.6	33.2	87.8	73.4	63.2
(v)	Without visual embedding	67.1	42.9	83.8	66.7	65.1
(vi)	Without auxiliary loss	71.8	47.0	85.8	69.4	68.5

Table 2. Mean average precision (mAP) performance of the proposed model (ii) compared to the baseline without query expansion (i) and to variations where parts of the model have been removed (iii-vi).

Self-attention: replacing the stack of self-attention encoders with a stack of fully-connected layers leads to a very noticeable drop in accuracy (iii), highlighting how important the attention is for this model.
Positional encoding (PE): Removing the PE (iv) leads to a very pronounced loss in accuracy for $\mathcal{R}$Oxford (which has very few relevant images per query). PE is necessary for queries with few relevant items because the model has to learn which images are important, and anchoring to the query (through the PE) enables it to do so. This is less important for queries with many relevant items, as in $\mathcal{R}$Paris. We additionally experiment with a position-only setup (v), where the self-attention computes the weights using only the positional encodings, not the actual image embeddings. This leads to a content-unaware weighting function, such as the AQE or AQEwD methods. The drop in accuracy is also remarkable, highlighting the need to combine both content and positional information.
Auxiliary loss: Removing the auxiliary loss (vi) leads to a small but consistent drop in accuracy. Although the model is fully functional without this auxiliary loss, it helps the optimization process to find better representations.
Inference time: When considering 64 neighbors for the expansion, our non-optimized PyTorch implementation can encode, on average, about 250 queries per second on a single Tesla M40 GPU. This does not include the time to extract the query embedding, which is orders of magnitude slower than our method (about 4 images per second on the same GPU) and the main bottleneck. Techniques such as distillation [38] and quantization [41], that have worked for transformer-based models, could further increase speed and reduce memory use.

4.4 Comparison with existing methods

Query expansion (QE). We compare the performance of our proposed method with existing QE approaches. All methods and their associated transformations are given in Table 1. For LAttQE, hyper-parameters are tuned on the validation set of rSfM120k, that has no overlapping landmarks or images with the test datasets. For competing methods, we select their hyper-parameters on the mean performance over test datasets, giving them an advantage. We denote the

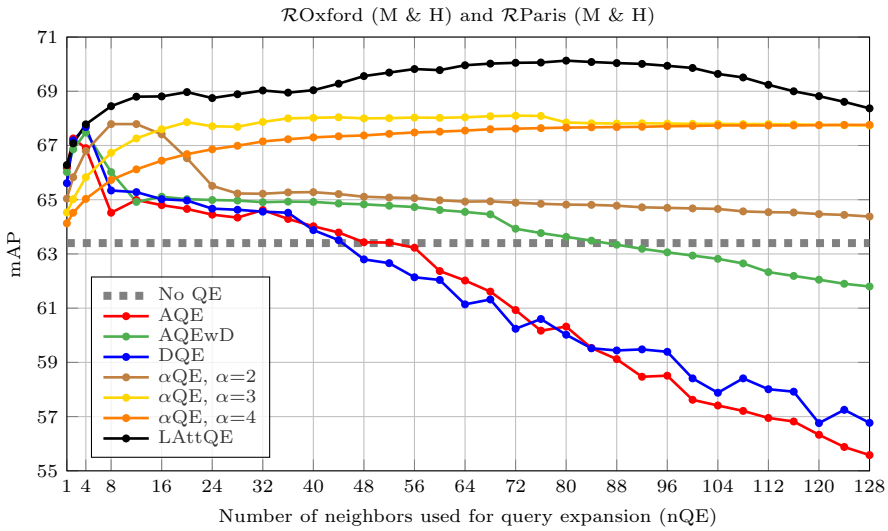

Fig. 3. Mean average precision over all queries of four protocols ($\mathcal{R}$Oxford (M & H) and $\mathcal{R}$Paris (M & H)) as a function of the number of neighbors used for query expansion.

number of neighbors used for QE as nQE. **AQE:** nQE=2; **AQEwD:** nQE=4; **αQE:** nQE=72, α=3; **DQE:** nQE=4, neg=5, $C = 0.1$; **LAttQE:** nQE=64.

Database-side augmentation (DBA). All of the before-mentioned methods can be combined with DBA. We separately tune all hyper-parameters in this combined scenario. We denote number of neighbors used for DBA as nDBA. **ADBA+AQE:** nDBA=4, nQE=4; **ADBAwD+AQEwD:** nDBA=4, nQE=6; **αDBA+αQE:** nDBA=36, nQE=10, α=3; **DDBA+DQE:** nDBA=4, nQE=2, C=0.1, neg=5; **LAttDBA+LAttQE:** nDBA=48, nQE=64.

Sensitivity to the number of neighbors used in the QE. Figure 3 shows the mean accuracy of LAttQE as well as other query expansion methods on $\mathcal{R}$Oxford and $\mathcal{R}$Paris, as a function of the number of neighbors used in the expansion. We highlight: (i) Unsurprisingly, methods that assume all samples are positive (*e.g.* AQE, DQE) degrade very fast when the number of neighbors is not trivially small. AQEwD degrades a bit more gracefully, but can still obtain very bad results if nQE is not chosen carefully. (ii) It is also unsurprising that αQE has become a standard, since the accuracy is high and results do not degrade when nQE is high. However, this only happens because of the weighting function is of the form r^α, with $r < 1$, *i.e.*, the weight rapidly converges to zero, and therefore most neighbors barely have any impact in the aggregation. (iii) Our proposed LAttQE consistently obtains the best results across the whole range of nQE. Our method is not limited by a weight that converges to zero, and therefore can still improve when αQE has essentially converged (nQE > 40).

Different "number of relevant images" and "AP" regimes. We evaluate query expansion impact at different regimes to showcase further differences

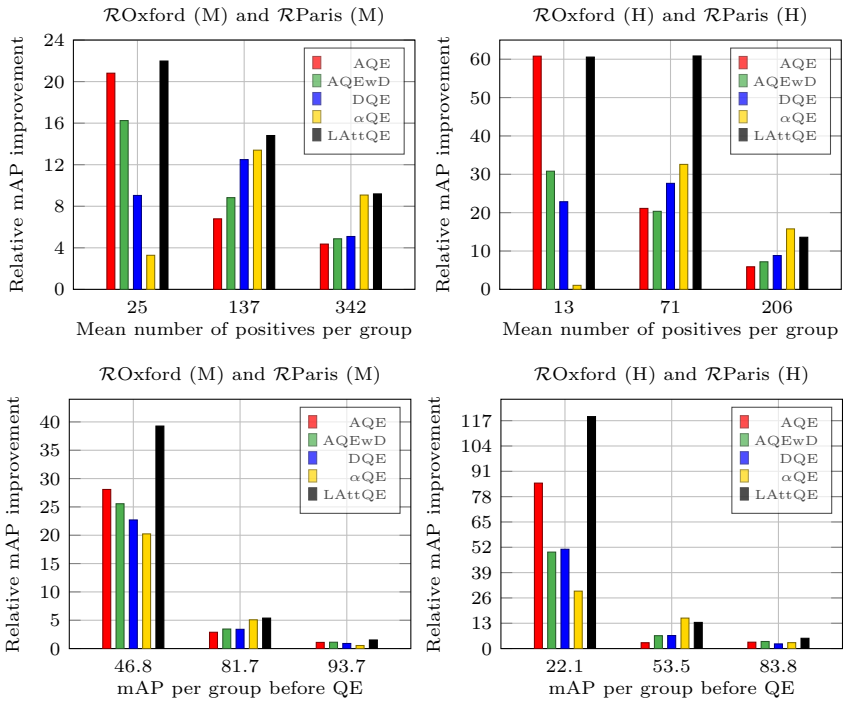

Fig. 4. Relative mean average precision (mAP) improvement at different number of relevant images (top) and AP regimes (bottom) split into 3 groups. Evaluation performed on $\mathcal{R}$Oxford and $\mathcal{R}$Paris at two difficulty setups, Medium (left) and Hard (right). Mean number of relevant images over all queries in the group (top) and mean average precision over all queries in the group (bottom) shown under respective group's bar plot.

between methods. In all cases we report the relative improvement in mAP introduced by using query expansion. In the first set of experiments, see Figure 4 (top), we group queries based on the number of relevant images, using percentiles 33 and 66 as cut-off. AQE (with nQE=4) works very well for queries with very few relevant samples, but leads to small improvements when the number of relevants is high, as they are not leveraged. On the other hand, αQE, with α=3 and nQE=72 obtains good results when the number of relevants is high, but really struggles when the number of relevants is low. LAttQE is the only method that is able to obtain high accuracy on all regimes. Figure 4 (bottom) groups queries based on their accuracy before query expansion. Similarly, LAttQE is the only method that consistently obtains high accuracy.

State-of-the-art comparison. Table 3 reports the accuracy of different methods on $\mathcal{R}$Oxford and $\mathcal{R}$Paris, both with and without the $\mathcal{R}$1M distractor set. The optimal number of neighbors for our approach (64 for LAttQE and 48 for LAttDBA) was decided on the validation set of rSfM120k. On the other hand, the optimal number of neighbors for the remaining methods was adjusted on test

	$\mathcal{R}$Oxf		$\mathcal{R}$Oxf + $\mathcal{R}$1M		$\mathcal{R}$Par		$\mathcal{R}$Par + $\mathcal{R}$1M		
	M	H	M	H	M	H	M	H	Mean
No QE									
—	67.3	44.3	49.5	25.7	80.6	61.5	57.3	29.8	52.0
QE									
[7] AQE	72.3	49.0	57.3	30.5	82.7	65.1	62.3	36.5	56.9
[13] AQEwD	72.0	48.7	56.9	30.0	83.3	65.9	63.0	37.1	57.1
[3] DQE	72.7	48.8	54.5	26.3	83.7	66.5	64.2	38.0	56.8
[33] αQE	69.3	44.5	52.5	26.1	**86.9**	**71.7**	66.5	41.6	57.4
$\star$ LAttQE	**73.4**	**49.6**	**58.3**	**31.0**	86.3	70.6	**67.3**	**42.4**	**59.8**
DBA + QE									
[7] ADBA + AQE	71.9	53.6	55.3	32.8	83.9	68.0	65.0	39.6	58.8
[13] ADBAwD + AQEwD	73.2	53.2	57.9	34.0	84.3	68.7	65.6	40.8	59.7
[3] DDBA + DQE	72.0	50.7	56.9	32.9	83.2	66.7	65.4	39.1	58.4
[33] αDBA + αQE	71.7	50.7	56.0	31.5	87.5	73.5	**70.6**	**48.5**	61.3
$\star$ LAttDBA + LAttQE	**74.0**	**54.1**	**60.0**	**36.3**	**87.8**	**74.1**	70.5	48.3	**63.1**

Table 3. Performance evaluation via mean average precision (mAP) on $\mathcal{R}$Oxford ($\mathcal{R}$Oxf) and $\mathcal{R}$Paris ($\mathcal{R}$Par) with and without 1 million distractors ($\mathcal{R}$1M). Our method is validated on validation part of rSfM120k and is marked with $\star$. Other methods are validated directly on mAP over all queries of 4 protocols of $\mathcal{R}$Oxford and $\mathcal{R}$Paris.

to maximize their mean accuracy on $\mathcal{R}$Oxford and $\mathcal{R}$Paris, giving them an unfair edge. Our method is the only one that consistently obtains good results on both $\mathcal{R}$Oxford and $\mathcal{R}$Paris. Compare this to other methods, where, for example, αQE obtains the best results on $\mathcal{R}$Paris but the worst results on $\mathcal{R}$Oxford, while AQE obtains the best results on $\mathcal{R}$Oxford (excepting our method) but the worst results on $\mathcal{R}$Paris. Generally, this gap becomes even larger when including the $\mathcal{R}$1M distractors. When using DBA and QE we observe the same trends: although some method can be slightly more accurate on specific datasets, our approach is the only one that obtains consistently good results on all datasets.

5 Conclusions

In this paper we have presented a novel framework to learn how to perform query expansion and database side augmentation for image retrieval tasks. Within this framework we have proposed LAttQE, an attention-based model that outperforms commonly used query expansion techniques on standard benchmark while being more robust on different regimes. Beyond LAttQE, we believe that the main idea of our method, tackling the aggregation for query expansion as a supervised task learned in a discriminative manner, is general and novel, and hope that more methods build on top of this idea, proposing new aggregation models that lead to more efficient and accurate search systems.

References

1. Alletto, S., Abati, D., Serra, G., Cucchiara, R.: Exploring architectural details through a wearable egocentric vision device. Sensors (2016) 1
2. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: CVPR (2016) 1
3. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR (2012) 2, 3, 4, 6, 8, 14
4. Azad, H.K., Deepak, A.: Query expansion techniques for information retrieval: a survey. IP&M (2019) 4
5. Chang, C., Yu, G., Liu, C., Volkovs, M.: Explore-exploit graph traversal for image retrieval. In: CVPR (2019) 4
6. Chum, O., Mikulík, A., Perdoch, M., Matas, J.: Total recall II: Query expansion revisited. In: CVPR (2011) 2, 3
7. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: CVPR (2007) 1, 2, 3, 4, 6, 14
8. Delvinioti, A., Jégou, H., Amsaleg, L., Houle, M.E.: Image retrieval with reciprocal and shared nearest neighbors. In: VISAPP (2014) 4
9. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019) 9
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019) 4
11. Fan, L., Zhao, H., Zhao, H., Liu, P., Hu, H.: Image retrieval based on learning to rank and multiple loss. IJGI (2019) 3
12. Girdhar, R., Ramanan, D.: Attentional pooling for action recognition. In: NeurIPS (2017) 4
13. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. IJCV (2017) 1, 2, 3, 4, 5, 6, 8, 14
14. Gu, Y., Li, C., Xie, J.: Attention-aware generalized mean pooling for image retrieval. arXiv:1811.00202 (2019) 3
15. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR (2006) 6
16. Heinly, J., Schonberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In: CVPR (2015) 1
17. Husain, S.S., Bober, M.: Remap: Multi-layer entropy-guided pooling of dense cnn features for image retrieval. TIP (2019) 3
18. 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. arXiv:1907.05794 (2019) 3
19. Iscen, A., Tolias, G., Avrithis, Y., Furon, T., Chum, O.: Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In: CVPR (2017) 4
20. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: ECCV (2008) 10
21. Kalantidis, Y., Tolias, G., Avrithis, Y., Phinikettos, M., Spyrou, E., Mylonas, P., Kollias, S.: Viral: Visual image retrieval and localization. Multimedia Tools and Applications (2011) 1
22. Lee, J., Lee, I., Kang, J.: Self-attention graph pooling. In: ICML (2019) 4

23. Liu, C., Yu, G., Volkovs, M., Chang, C., Rai, H., Ma, J., Gorti, S.K.: Guided similarity separation for image retrieval. In: NIPS (2019) 4
24. Makantasis, K., Doulamis, A., Doulamis, N., Ioannides, M.: In the wild image retrieval and clustering for 3d cultural heritage landmarks reconstruction. Multimedia Tools and Applications (2016) 1
25. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008) 4
26. Maron, M.E., Kuhns, J.L.: On relevance, probabilistic indexing and information retrieval. JACM (1960) 4
27. Mikulik, A., Chum, O., Matas, J.: Image retrieval for online browsing in large image collections. In: SISAP (2013) 1
28. Ng, T., Balntas, V., Tian, Y., Mikolajczyk, K.: Solar: Second-order loss and attention for image retrieval. arXiv:2001.08972 (2020) 9
29. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: ICCV (2017) 2, 9
30. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007) 9, 10
31. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR (2008) 10
32. Qin, D., Gammeter, S., Bossard, L., Quack, T., Van Gool, L.: Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In: CVPR (2011) 4
33. Radenovic, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. TPAMI (2018) 1, 2, 3, 4, 5, 6, 9, 10, 14
34. Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting oxford and paris: Large-scale image retrieval benchmarking. In: CVPR (2018) 3, 10
35. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog (2019) 4
36. Revaud, J., Almazan, J., de Rezende, R.S., de Souza, C.R.: Learning with average precision: Training image retrieval with a listwise loss. In: ICCV (2019) 3, 5, 9
37. Rocchio, J.: Relevance feedback in information retrieval. The SMART Retrieval System (1971) 4
38. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: NeurIPS Workshop (2019) 11
39. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image retrieval for image-based localization revisited. In: BMVC (2012) 1
40. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 1
41. Shen, S., Dong, Z., Ye, J., Ma, L., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Q-BERT: Hessian based ultra low precision quantization of BERT. In: AAAI (2020) 11
42. Shen, X., Lin, Z., Brandt, J., Wu, Y.: Spatially-constrained similarity measurefor large-scale object retrieval. TPAMI (2013) 4
43. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV (2003) 1, 3
44. Tolias, G., Avrithis, Y., Jégou, H.: Image search with selective match kernels: aggregation across single and multiple images. IJCV (2015) 8
45. Tolias, G., Jégou, H.: Visual query expansion with or without geometry: refining local descriptors by feature aggregation. PR (2014) 2, 3
46. Turcot, T., Lowe, D.G.: Better matching with fewer features: The selection of useful features in large database recognition problems. In: ICCV Workshop (2009) 8

47. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Łukasz Kaiser, Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 4, 6, 7
48. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018) 4
49. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. JMLR (2009) 6
50. Weyand, T., Leibe, B.: Discovering favorite views of popular places with iconoid shift. In: ICCV (2011) 1